



## SPIS TREŚCI NUMERU 11 (402)

Bajka o Kocie w Butach, czyli jak zamienić człowieka w mysz <i>Anna Gambin</i>	str. 1
Problem RMQ <i>Jakub Radoszewski</i>	str. 4
 Empiryczne badanie uogólnionej wersji gry Hamleta <i>Andrzej Walat</i>	str. 6
Informatyczny kącik olimpijski (2) <i>Filip Wolski</i>	str.10
<b>OLIMPIADA INFORMATYCZNA</b>	str.11
 Ilu liderów? <i>Michał Adamaszek</i>	str.15
 O pożytkach z siatki <i>Marek Kordos</i>	str.16
Czarownica w nieskończonym lesie <i>Marcin Pilipczuk</i>	str.17
Euler, Lagrange i trzy ciała <i>Mikołaj Korzyński</i>	str.18
Limeryki <i>Przemysław Kiciak</i>	str.19
 Badamy siłę elektrodynamiczną <i>Stanisław Bednarek</i>	str.20
 Zadania	str.21
Aktualności	str.22
Klub 44	str.23
Można też inaczej <i>Michał Adamaszek</i>	str.23
Patrz w niebo	str.24
Listopad	str.24
 Losowe rozbitcia odcinka, proces Poissona i paradoks czasu oczekiwania <i>Rafał Sztencel</i>	str.25

**W następnym numerze:**  
**Bohdan Paczyński o astronomii**

Miesięcznik *Delta* – matematyka, fizyka, astronomia, informatyka jest wydawany przez Uniwersytet Warszawski przy współpracy towarzystw naukowych: Polskiego Towarzystwa Matematycznego, Polskiego Towarzystwa Fizycznego i Polskiego Towarzystwa Astronomicznego.

Komitet Redakcyjny: prof. dr hab. Andrzej Białynicki-Birula (członek rzeczywisty PAN), prof. dr hab. Bogdan Cichocki, dr Krzysztof Ciesielski – wiceprzewodniczący, dr hab. Armen Edigarian, prof. dr hab. Jan A. Gaj – przewodniczący, dr hab. Maciej Geller, prof. dr hab. Jerzy Ginter, dr Piotr Goldstein, dr Agnieszka Janiuk, prof. dr hab. Wiesław A. Kamiński, dr hab. Andrzej Majhofer, dr hab. Zbigniew Marciniak, prof. dr hab. Janusz Matkowski, dr Adam Michalec, prof. dr hab. Ryszard J. Pawlak, dr Zdzisław Pogoda, prof. dr hab. Grzegorz Sitarski, dr Weronika Śliwa, prof. dr hab. Andrzej Woszczyk.

Redaguje kolegium w składzie: Michał Adamaszek, Wiktor Bartol, Ewa Czuchry, Marcin Hauzer, Krystyna Kordos – sekr. red., Marek Kordos – red. nac., Mikołaj Korzyński, Tomasz Kwast, Urszula Marciniak, Anna Rudnik, Witold Sadowski, Andrzej Walat, Piotr Zalewski – z-ca red. nac. Okładki i rysunki: Emilia Bojańczyk. Rysunki techniczne: Marcin Adamski.

Adres do korespondencji:  
Instytut Matematyki UW, Redakcja „Delta”, ul. Banacha 2, pokój 5450, 02-097 Warszawa, e-mail: delta@mimuw.edu.pl, tel. 022-55-44-545.

Skład systemem  $\text{T}_{\text{E}}\text{X}$  wykonała Redakcja.

Wydrukowano w Drukarni Greg, ul. Konstruktorska 4, 02-673 Warszawa.

### WARUNKI PRENUMERATY W FIRMIE AMOS

01-785 Warszawa, ul. Broniewskiego 8A (tel. 022-663-87-52, 022-663-11-46)  
internet: www.amos.waw.pl, e-mail: biuro@amos.waw.pl

Wpłaty przyjmowane są non-stop, do 10. dnia miesiąca poprzedzającego okres prenumeraty. **Okres prenumeraty wynosi co najmniej trzy miesiące.** Cena jednego numeru w 2008 roku wynosi 4 zł. Przy wpłacie prosimy o zaznaczenie okresu prenumeraty.

W prenumeracie zagranicznej (też przez okres **co najmniej trzech miesięcy**) cena numeru w 2008 r. wynosi 8 zł. W przypadku życzenia dostawy priorytetowej odpowiednią dopłatę ponosi zamawiający.

**Uwaga!** Dla zamawiających minimum 10 egzemplarzy każdego numeru AMOS funduje dodatkowo jeden egzemplarz pisma.

Konto AMOS-u: PKO BP SA I O/W-wa, nr 11 1020 1013 0000 0502 0004 0584

### WARUNKI PRENUMERATY W RUCH-u

internet www.ruch.com.pl, infolinia 0-804-200-600

Cena prenumeraty w 2008 roku wynosi 4 zł za egzemplarz.

1. **Prenumerata krajowa:** wpłaty przyjmują jednostki kolportażowe „RUCH” SA właściwe dla miejsca zamieszkania. Termin przyjmowania prenumeraty: do 5 każdego miesiąca poprzedzającego okres rozpoczęcia prenumeraty.

2. **Prenumerata ze zleceniem wysyłki za granicę:** informacji o warunkach prenumeraty i sposobie zamawiania udziela „RUCH” SA Oddział Krajowej Dystrybucji Prasy, 01-248 Warszawa, ul. Jana Kazimierza 31/33; tel. 022-5328-731 (prenumerata płatna w walucie obcej), -816, -734, -819 (prenumerata płatna w PLN w kasie Oddziału lub na konto w banku PEKAO SA IV O/Warszawa 68 1240 1053 1111 0000 0443 0494), infolinia 0-800-1200-29.

3. **Prenumerata opłacana za granicą:** przelewem na nasze konto SWIFT banku: PKOPPLPWWA4; w USD PEKAO SA IV O/W-wa IBAN PL54 1240 1053 1787 0000 0443 0508; w EUR PEKAO SA IV O/W-wa IBAN PL54 1240 1053 1978 0000 0443 0511; kserokopię polecenia przelewu należy przesłać faksem pod numer +48-22-5328-731.

Numery archiwalne (od 1987 r.) można nabyć w Redakcji osobiście lub listownie.

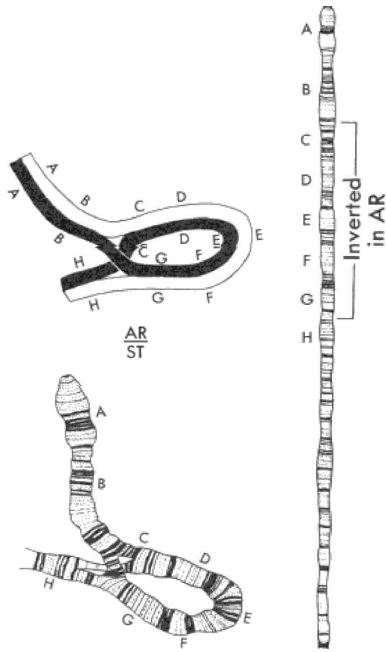
Strona internetowa (streszczenia, artykuły archiwalne, linki itd.):  
<http://www.mimuw.edu.pl/delta>

Wydawca: Uniwersytet Warszawski

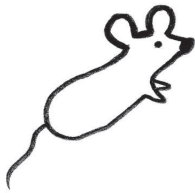
Cena 1 egzemplarza 4 zł

# Bajka o Kocie w Butach, czyli jak zamienić człowieka w mysz

Anna GAMBIN\*



Rys 1. Tak zdefiniowany rewersal modeluje zjawisko molekularne zaobserwowane już w 1938 roku u muszki owocówki. Na rysunku widzimy prawdziwy rewersal, który polega na tym, że fragment chromosomu zawija się w pętelkę, a następnie pętelka ta zostaje powtórnie rozciągnięta w taki sposób, że kolejność genów położonych wewnątrz ulega odwróceniu.



**Naleśniki Billa Gatesa.** Problem sortowania przez rewersale z dodatkowym ograniczeniem, że używamy tylko rewersali typu  $\rho(1, i)$ , czyli tzw. rewersali prefiksowych, był badany przez studenta Harvardu Billa Gatesa i jego promotora Cristosa Papadimitriou. Podali oni ograniczenia na liczbę  $d_{pref}(n) = \max_{\pi \in S_n} d_{pref}(\pi)$ .  $S_n$  oznacza tutaj zbiór wszystkich permutacji  $n$ -elementowych. Motywacją do sortowania przez rewersale prefiksowe nie dostarczyła biologia molekularna, lecz następująca zagadka kulinarna: wyobraźmy sobie niestarannego kucharza, który smaży naleśniki różnej wielkości oraz kelnera pedanta, który pragnie podać gościom na stół tacę z naleśnikami ułożonymi w zgrabną piramidę – od największego na dole do najmniejszego na górze.  $d_{pref}(\pi)$  jest minimalną liczbą operacji odwrócenia czubka naleśnikowej góry, jaką musi wykonać kelner, żeby posortować naleśniki. Znane oszacowania są następujące:  $\frac{17}{16}n \leq d_{pref}(n) \leq \frac{5}{3}n + \frac{5}{3}$ . Problem znalezienia dokładnej wartości  $d_{pref}(n)$  nadal czeka na rozwiązanie.

\*Instytut Informatyki, Uniwersytet Warszawski

Większość z nas zaakceptowała fakty, do których już od połowy dziewiętnastego stulecia przekonywał nas Karol Darwin i jego następcy, mówiące, że człowieka łączy wyjątkowo bliskie pokrewieństwo z małpami. Dużo trudniej zgodzić się z tym, że gatunek ludzki bardzo wiele wspólnego ma z pospolitymi myszkami – tutaj różnice na pierwszy rzut oka są jeszcze bardziej ewidentne. Okazuje się jednak, że genetycznie jesteśmy bardzo podobni. Jeżeli wyobrazimy sobie genom myszy (20 par chromosomów) jako długi naszyjnik nawleczonych na nitkę genów, to wystarczy pociąć go na około 200 fragmentów i odpowiednio powiązać, aby otrzymać 23 pary chromosomów człowieka. Oznacza to, że stosunkowo niewiele *rearanżacji genomowych* miało miejsce na przestrzeni, bagatela, 80 milionów lat, kiedy to nasza ewolucja podążała odmiennymi torami. Dla ścisłości wyjaśniam, że zaniedbujemy w tym podejściu dużo częstsze zmiany genomu, takie jak mutacje punktowe i uznajemy dwa fragmenty DNA za równoważne, jeśli kodują geny pełniące w rozważanych organizmach taką samą funkcję.

Na pewno pamiętacie bajkę o Kocie w Butach i jego wielkim sukcesie, kiedy namówił złego czarownika do przybrania postaci polnej myszki. Po tym ruchu wystarczyło szybko myszkę połknąć, a rozległe dobra czarownika, w tym przepiękny pałac, stały się własnością Jasia i jego szlachetnie urodzonej małżonki.

Zostaniemy teraz czarownikiem i zamienimy człowieka w mysz za pomocą genetycznych manipulacji, starając się, aby uczynić ten proces jak najbardziej efektywnym. Jak się domyślicie, będzie nam potrzebny matematyczny model zjawiska rearanżacji genomu. Na początek założmy, że genom będzie reprezentowany przez permutację  $\pi = (\pi_1, \pi_2, \dots, \pi_n)$  zbioru  $\{1, 2, \dots, n\}$ . O elementach permutacji możemy myśleć jako o genach albo, co jest bliższe prawdy, większych spójnych fragmentach chromosomu. Do tasowania porządku genów użyjemy operacji o nazwie *rewersal* (oznaczymy ją literką  $\rho$ ). Operacja ta ma dwa parametry,  $i$  oraz  $j$ , a zastosowana do permutacji  $\pi = (\pi_1, \pi_2, \dots, \pi_n)$ , odwraca porządek genów od  $i$ -tego do  $j$ -tego, czyli:

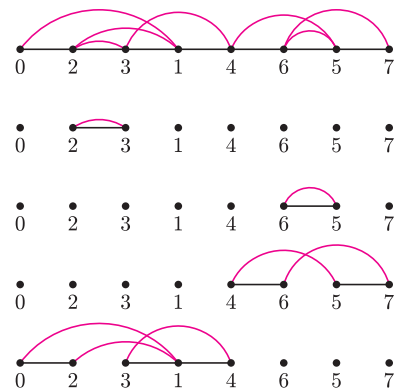
$$\rho(i, j) = (\pi_1, \dots, \pi_{i-1}, \pi_j, \pi_{j-1}, \dots, \pi_{i+1}, \pi_i, \pi_{j+1}, \dots, \pi_n).$$

Nasze efektywne czary polegają na przekształceniu permutacji  $\pi$  (człowiek) w permutację  $\delta$  (myszka) przy użyciu jak najmniejszej liczby rewersali. Tak więc poszukujemy ciągu takich operacji  $\rho_1, \rho_2, \dots, \rho_t$ , że  $\rho_t \dots \rho_2 \rho_1 \pi = \delta$  oraz  $t$  jest jak najmniejsze. Liczbę  $t$  nazwiemy odległością rewersalową pomiędzy permutacjami  $\pi$  i  $\delta$ . Nie jest trudno zauważyć, że możemy bez straty ogólności rozważać problem *sortowania przez rewersale*, czyli znalezienia odległości  $d(\pi)$  permutacji  $\pi$  od permutacji identyecznościowej  $(1, 2, \dots, n)$ .

Okazuje się, że posortowanie permutacji przy użyciu rewersali jest zadaniem dość skomplikowanym. Jeśli zajmiemy się jedynie chromosomem X myszy i człowieka (ten chromosom zawiera u większości ssaków bardzo podobne geny i może być reprezentowany jako permutacja długości 7), to stosunkowo nietrudno zweryfikować fakt, że odległość rewersalowa pomiędzy nimi wynosi 6. Sytuacja komplikuje się bardzo, kiedy rozważamy genomy o większej liczbie bloków zachowujących porządek genów, czyli dłuższe permutacje.

Do oszacowania  $d(\pi)$  bardzo pomocne okazuje się pojęcie *punktu złamania* permutacji (ang. *breakpoint*). Przyjmijmy oznaczenie  $i \sim j$ , jeśli  $|i - j| = 1$ . Dodatkowo na obydwu końcach permutacji  $\pi = (\pi_1, \pi_2, \dots, \pi_n)$  dopiszmy  $\pi_0 = 0$  oraz  $\pi_{n+1} = n + 1$ . Parę kolejnych elementów permutacji  $(\pi_i, \pi_{i+1})$  dla  $0 \leq i \leq n$  nazwiemy *sąsiedztwem*, jeśli  $\pi_i \sim \pi_{i+1}$  czyli są to po prostu kolejne liczby. Przez punkt złamania permutacji będziemy z kolei rozumieć parę  $(\pi_i, \pi_{i+1})$ , dla której  $\pi_i \not\sim \pi_{i+1}$ . Zauważmy, że permutacja identyecznościowa nie ma żadnego punktu

Oczywiście, nie można wykluczyć, że ktoś rozwiąże wartą milion dolarów zagadkę, pokazując, iż klasa problemów wielomianowych jest tożsama z klasą problemów niedeterministycznych wielomianowych, czyli  $P = NP$ . Więcej o zagadnieniach złożoności obliczeniowej możecie przeczytać w numerze *Delta* 01/2007.



Rys. 2. Graf złamań i jego dekompozycja na cykle dla permutacji  $\pi = (2, 3, 1, 4, 6, 5)$ .

złamania, czyli proces sortowania polega na eliminacji punktów złamania.

Dowolny rewersal potrafi wyeliminować co najwyżej dwa punkty złamania, co pozwala uzyskać oszacowanie:  $d(\pi) \geq \frac{b(\pi)}{2}$ , gdzie  $b(\pi)$  oznacza liczbę punktów złamania permutacji  $\pi$ .

Korzystając z powyższego oszacowania, zaproponowano wielomianowy algorytm aproksymacyjny ze współczynnikiem aproksymacji 2 dla problemu sortowania przez rewersale. Oznacza to, że algorytm wygeneruje ciąg rewersali co najwyżej dwa razy dłuższy niż optymalny. Udowodniono też, że problem sortowania przez rewersale jest  $NP$ -trudny, czyli najprawdopodobniej nie istnieje algorytm o złożoności wielomianowej sortujący permutację za pomocą minimalnej liczby rewersali.

Ponieważ oszacowanie  $d(\pi)$  za pomocą liczby punktów złamania jest najczęściej bardzo niedokładne, zaproponowano kolejne pojęcia mające elegancką, grafową interpretację. Dla permutacji  $\pi$  skonstruujemy *graf złamań*  $G(\pi)$ , który będzie miał  $n + 2$  wierzchołki etykietowane liczbami  $\{0, 1, 2, \dots, n, n + 1\}$ . Każde dwa wierzchołki  $(\pi_i, \pi_{i+1})$  łączymy krawędzią w kolorze czarnym, natomiast parę  $(\pi_i, \pi_j)$  łączymy kolorową krawędzią, o ile  $\pi_i \sim \pi_j$ , czyli są to kolejne liczby w permutacji identycznościowej. Konstrukcję grafu możemy wyobrazić sobie jako nawlekanie na czarną nitkę kolejnych genów jednego organizmu, a następnie nawlekanie na kolorową nić kolejnych genów drugiego organizmu. Cykl w grafie to taka ścieżka, która zaczyna się i kończy w tym samym wierzchołku. Cykl nazwiemy *alternującym*, jeśli kolory kolejnych krawędzi na naszej ścieżce będą zmieniać się naprzemiennie.

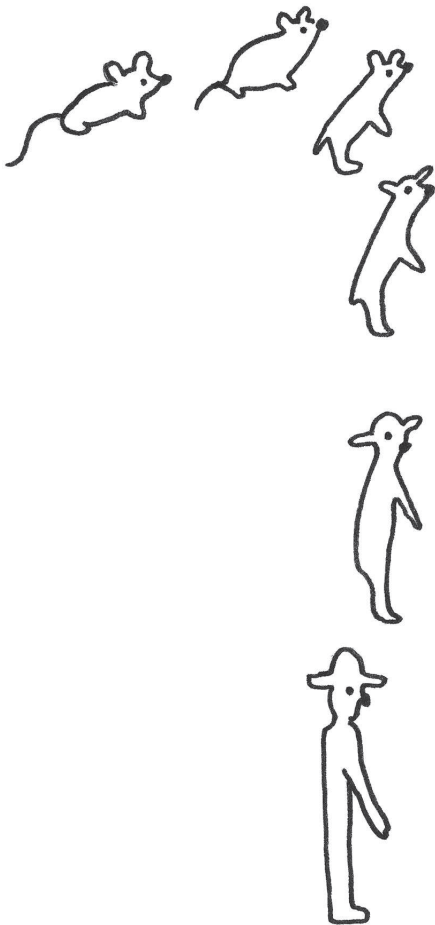
Nasz dwukolorowy graf ma ciekawą własność nazywaną *zbalansowaniem*.

Oznacza to, że każdy jego wierzchołek jest zbalansowany, czyli wychodzi z niego taka sama liczba krawędzi kolorowych co czarnych. Są to zawsze dwie czarne krawędzie i dwie kolorowe, z wyjątkiem dwóch skrajnych elementów permutacji, czyli 0 i  $n + 1$ . Nietrudno pokazać, że grafy zbalansowane mają tzw. *cykl Eulera*, czyli cykl, który zawiera wszystkie krawędzie w grafie i każdą przechodzi dokładnie raz.

Z faktu, że graf złamań dla permutacji ma *alternujący cykl Eulera*, wnioskujemy, iż można przyporządkować wszystkie krawędzie grafu rozłącznym cykлом alternującym w taki sposób, że każda krawędź występuje w dokładnie jednym cyklu. Nazywamy taką operację *dekompozycją na cykle* grafu  $G(\pi)$ . Graf z rysunku 2 może zostać zdekomponowany na 4 cykle alternujące:  $2 \rightarrow 3 \rightarrow 2$ ,  $6 \rightarrow 5 \rightarrow 6$ ,  $4 \rightarrow 5 \rightarrow 7 \rightarrow 6 \rightarrow 4$  oraz  $0 \rightarrow 1 \rightarrow 3 \rightarrow 4 \rightarrow 1 \rightarrow 2 \rightarrow 0$ . Okazuje się, że więcej cykli nie uzyskamy w żadnej innej dekompozycji tego grafu, czyli jest to dekompozycja na maksymalną liczbę cykli – oznaczmy tę liczbę przez  $c(\pi)$ .

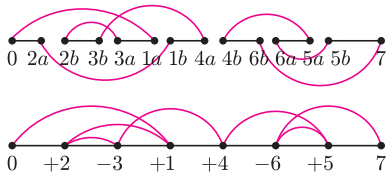
Pożyteczna dla nas własność dekompozycji na cykle jest następująca: kiedy zastosujemy dowolny rewersal do permutacji  $\pi$ , liczba  $c(\pi)$  może się zmienić co najwyżej o jeden. Oznacza to po prostu, że wykonując rewersal w celu posortowania permutacji, możemy dodać co najwyżej jeden cykl. Zauważmy, że graf złamań permutacji identycznościowej jest zdekomponowalny na maksymalną liczbę cykli alternujących, czyli  $n + 1$ . Udowodniono, że zachodzi oszacowanie  $d(\pi) \geq n + 1 - c(\pi)$ , które jest o wiele bardziej dokładne, niż rozważane poprzednio  $d(\pi) \geq b(\pi)/2$  (gdzie, jak pamiętamy,  $b(\pi)$  oznacza liczbę punktów złamania permutacji  $\pi$ ).

Niestety, nie zawsze wykonanie rewersalu doda jeden cykl do naszej dekompozycji na maksymalną liczbę cykli alternujących. Dlatego za pomocą liczby cykli potrafimy jedynie oszacować odległość rewersalową. Dobra wiadomość jest taka, że dla permutacji o molekularnym rodowodzie (to znaczy takich, które modelują porządek genów na chromosomie) w powyższym cyklowym oszacowaniu zachodzi najczęściej równość, czyli  $d(\pi) = n + 1 - c(\pi)$ . Wydawałoby się, że ta obserwacja powinna zagwarantować sukces naszym czarom: wystarczy znaleźć dekompozycję grafu złamań  $G(\pi)$  na maksymalną liczbę rozłącznych krawędziowo cykli alternujących  $c(\pi)$ , policzyć te cykle i wykonać odpowiednie rewersale, a z chromosomu złego czarownika dostaniemy





chromosom małej myszki. Okazuje się jednak, że nasze czarodziejskie matematyczne zaplecze jest nadal za słabe, bo problem znalezienia wspomnianej dekompozycji jest złożonościowo zbyt trudny.



Rys. 3. Graf złamań i jego jednoznaczna dekompozycja na cykle dla permutacji ze znakami  $\pi = (+2, -3, +1, +4, -6, +5)$ .

Na szczęście w sukurs przychodzi nam znów biologia – przypomnijmy sobie, że nić DNA jest obiektem skierowanym (ma swój początek – przez biologów zwany 5' i koniec ochrzczoney 3' – obydwie nazwy pochodzą od położenia atomów węgla w cząsteczce cukru deoksyrybozy). Podobnie geny leżące na nici DNA są skierowane, czyli powinniśmy nasz permutacyjny model organizmu bardziej skomplikować. Komplikacja nie będzie wielka, natomiast zysk z lepszego modelu znaczący. Porządek genów będziemy teraz reprezentować jako *permutację ze znakami*, czyli z każdym elementem wiążemy znak '+' lub '-', wskazujący, czy gen jest skierowany w prawo czy też w lewo. Oczywiście, geny leżące na jednej nici DNA są skierowane w tę samą stronę, ale podwójna spirala DNA składa się z dwóch nici DNA biegnących w przeciwnych kierunkach – stąd różne znaki w naszej permutacji odpowiadają temu, że gen pochodzi z jednej bądź z drugiej nici DNA.



Zastanówmy się teraz, jak działa rewersal  $\rho(i, j)$  na permutacji ze znakami. Odwraca on na pewno porządek genów, ale też zamienia ich znaki (kierunki). Dla przykładu, stosując rewersal  $\rho(2, 5)$  do permutacji  $(+1, -5, -4, -3, -2)$ , dostaniemy skierowaną permutację identycznościową  $(+1, +2, +3, +4, +5)$ . Rewersal  $\rho(i, i)$  zamienia znak genu  $i$ . Znowu interesuje nas minimalna liczba rewersali  $d(\pi)$ , które przekształcą permutację ze znakami  $\pi$  w skierowaną permutację identycznościową  $(+1, +2, \dots, +n)$ . Dla permutacji ze znakami możemy też zbudować graf złamań, zastępując każdy wierzchołek w grafie  $G(\pi)$  przez dwa wierzchołki symbolizujące początek i koniec genu. Oznaczmy te wierzchołki dla genu  $i$  jako  $ia$  (początek) oraz  $ib$  (koniec). Tak więc gen  $+2$  będzie reprezentowany przez parę sąsiednich wierzchołków  $2a$  i  $2b$ , natomiast gen  $-3$  będzie reprezentowany jako para  $3b$  i  $3a$ . Czarnymi krawędziami łączymy teraz sąsiednie końce genów, czyli dla permutacji z rysunku 3

$$\pi = (+2, -3, +1, +4, -6, +5)$$

łączymy wierzchołki  $2b$  i  $3b$ , następnie  $3a$  i  $1a$ , itd. Kolorowe krawędzie połączą, tak jak poprzednio, geny sąsiednie w permutacji identycznościowej (skierowanej), czyli wierzchołek  $1b$  łączymy z  $2a$ ,  $2b$  z  $3a$ ,  $3b$  z  $4a$ , itd.

W tej chwili wystarczy, że spojrzycie na obrazek na marginesie, a dokonacie zaskakującego odkrycia: w naszym nowym grafie dekompozycja na alternujące cykle jest jednoznaczna! Ma ona w szczególności maksymalną liczbę cykli. W naszym przykładzie są dwa takie cykle:  $0 \rightarrow 1a \rightarrow 3a \rightarrow 2b \rightarrow 3b \rightarrow 4a \rightarrow 1b \rightarrow 2a \rightarrow 0$  oraz  $4b \rightarrow 5a \rightarrow 6a \rightarrow 5b \rightarrow 7 \rightarrow 6b \rightarrow 4b$ . Wnioskujemy stąd, że aby posortować naszą przykładową permutację, musimy użyć co najmniej  $d(\pi) = n + 1 - c(\pi) = 6 + 1 - 2 = 5$  rewersali. Okazuje się, że 5 rewersali wystarczy (spróbujcie!).

Powinniśmy właściwie zakończyć teraz molekularno-matematyczne czary usatysfakcjonowani faktem, że większość złych czarowników potrafimy efektywnie zamieniać w wiele niegroźnych zwierzaków. Warto jeszcze dodać, że uparci matematycy nie dali za wygraną i znaleźli dla permutacji ze znakiem dokładną liczbę rewersali niezbędną do posortowania. Wynosi ona  $d(\pi) = n + 1 - c(\pi) + h(\pi) + f(\pi)$ , gdzie  $c(\pi)$  jest znaną nam liczbą cykli alternujących,  $h(\pi)$  jest liczbą *plotków* w permutacji, natomiast  $f(\pi)$  wynosi 1, kiedy permutacja  $\pi$  jest *fortecą* (jeśli  $\pi$  nie jest *fortecą*, to  $f(\pi) = 0$ ). Nazwy *plotek* i *forteca* sugerują przeszkody, które trzeba pokonać, sortując permutację. Ich zdefiniowanie, niestety, wymagałoby o wiele dłuższych rozważań.

Na zakończenie wspomnijmy, że matematyczna teoria sortowania przez rewersale pozwoliła dokonać ciekawego biologicznego odkrycia: rozważając najbardziej efektywny scenariusz transformacji jednego organizmu w drugi, spotykamy rewersale, które bardzo często są zaczepione w ustalonym punkcie w chromosomie. Własności takich „wrażliwych” miejsc, odkrytych przez matematyków, są teraz badane przez biologów.



### Rozwiązanie zadania F 703.

Załóżmy, że ciśnienie wewnątrz cylindrów zwiększyło się do wielkości  $p$ . Wtedy tłoki zostały przesunięte w lewo na odległość  $d$ . Z prawa Boyle'a-Mariotte'a mamy, że:

$$p_0 V = p(V - S_1 d + S_2 d),$$

stąd

$$d = \frac{(1 - p_0/p)V}{S_1 - S_2}.$$

W całym artykule  $\log n$  oznacza logarytm o podstawie 2.

## Problem RMQ

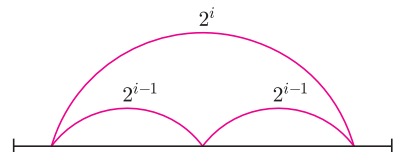
Jakub RADOSZEWSKI\*

**Przypomnienie.** W *Delcie* 9/2007 zajmowaliśmy się dwoma problemami: RMQ i LCA. Przypomnijmy, że problem LCA (*Lowest Common Ancestor*) polega na zbudowaniu struktury danych, która dla danego drzewa pozwoli szybko (w czasie  $O(1)$ ) odpowiadać na zapytania o najniższego wspólnego przodka danych dwóch wierzchołków. Z kolei w problemie RMQ (*Range Minimum Query*) poszukujemy takiej struktury, która pozwoli dla danego ciągu  $a_1, \dots, a_n$  szybko odpowiadać na zapytania  $RMQ(i, j)$  o minimalny element fragmentu ciągu  $a_i, a_{i+1}, \dots, a_{j-1}, a_j$ , wyznaczonego przez  $i$ -ty i  $j$ -ty jego wyraz.

Udowodniliśmy, że te dwa problemy są równoważne, a dowód polegał na pokazaniu liniowego sprowadzenia problemu RMQ do LCA dla pewnego drzewa, a następnie problemu LCA do RMQ dla pewnego ciągu. W tym numerze – zgodnie z zapowiedzią – zaprezentujemy efektywne rozwiązanie problemu RMQ, które na mocy powyższych rozważań będzie jednocześnie rozwiązaniem problemu LCA. W celu uproszczenia dalszego wywodu będziemy zakładać, że długość analizowanego ciągu ( $n$ ) jest potęgą dwójki; nie jest to duże ograniczenie, gdyż w przeciwnym przypadku możemy ciąg uzupełnić, na przykład, zerami do uzyskania odpowiedniej długości, co nie spowoduje nadmiernego jego wydłużenia.

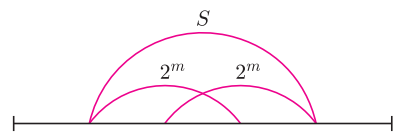
**Pierwsze podejście.** Na początek skupimy się na spójnych fragmentach ciągu  $a_1, \dots, a_n$  (będziemy je odtąd nazywać *segmentami*), których liczba elementów jest potęgą dwójki i spróbujemy dla każdego z nich wyznaczyć minimalny element. Zastanówmy się najpierw, ile jest takich segmentów w naszym ciągu. Liczba różnych potęg dwójki nie większych od  $n$  jest rzędu  $\log n$ , a dla każdej długości mamy co najwyżej  $n$  odpowiadających jej segmentów, co daje łączną liczbę  $O(n \log n)$ .

Segmentów długości 1 mamy  $n$  i wynik dla każdego z nich to po prostu zadany element ( $RMQ(i, i) = a_i$ ). Każdy segment długości 2 składa się z dwóch sąsiadujących ze sobą segmentów długości 1, więc dla każdego z nich jako minimum wybieramy mniejsze z minimów odpowiednich segmentów długości 1 ( $RMQ(i, i+1) = \min(RMQ(i, i), RMQ(i+1, i+1))$ ). To postępowanie kontynuujemy dla kolejnych długości: minimum z segmentu długości  $2^i$  liczymy jako minimum z minimów segmentów długości  $2^{i-1}$ , które się składają na rozważany większy segment (rys. 1). Ostatecznie dla każdego z rozważanych segmentów obliczyliśmy wynik w czasie stałym, co daje łączną złożoność tej fazy algorytmu równą  $O(n \log n)$ .



Rys. 1

Po wykonaniu wstępnych obliczeń możemy teraz udzielać odpowiedzi na zapytania o  $RMQ$  dowolnego segmentu  $S$  ciągu. Wystarczy znaleźć największą wartość potęgi dwójki  $2^m$ , nie większą od długości rozważanego przedziału (złożoność czasowa  $O(\log n)$ ), a następnie jako wynik przyjąć minimum z wyników dwóch nachodzących się segmentów długości  $2^m$ , które pokrywają cały rozważany segment (rys. 2). Możemy dla każdej długości segmentu od 1 do  $n$  obliczyć na wstępie największą wartość potęgi dwójki nie większą od niej w łącznej złożoności czasowej  $O(n \log n)$ . W ten sposób złożoność czasowa wstępnych obliczeń w takim algorytmie będzie  $O(n \log n)$ , a każde zapytanie będziemy mogli obsłużyć w złożoności czasowej  $O(1)$ .



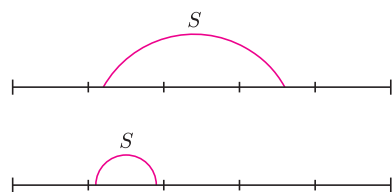
Rys. 2

Otrzymany algorytm jest bardzo szybki i wystarczający dla większości praktycznych zastosowań. My jednak zajmiemy się jego doskonaleniem, dążąc do algorytmu, wykonującego jedynie  $O(n)$  wstępnych obliczeń i obsługującego zapytania w czasie stałym.

**Właściwy algorytm.** W celu rozwiązania problemu RMQ sprowadzamy go w liniowej złożoności czasowej do problemu LCA. Następnie, również w liniowej złożoności czasowej, wykonujemy sprowadzenie otrzymanego LCA z powrotem do RMQ. Nasze postępowanie z pozoru wydaje się nie przynosić żadnych

\*student, Wydział Matematyki, Informatyki i Mechaniki, Uniwersytet Warszawski

korzyści, gdyż w jego efekcie znajdujemy się w punkcie wyjścia – przeszliśmy z jednego problemu RMQ do innego. Są to jednak tylko pozory: końcowe zagadnienie RMQ nie jest wszakże tym, od którego rozpoczynaliśmy, a ma jedną bardzo istotną własność: w otrzymanym ciągu różnica kolejnych dwóch elementów wynosi 1 albo  $-1$ . Wynika to z faktu, że wynikowy ciąg składa się z wysokości kolejno odwiedzanych w algorytmie przeszukiwania węzłów drzewa. Z kolei w żadnym kroku przeszukiwania nie występują „przeskoki” – za każdym razem z danego wierzchołka przechodzimy albo do jego ojca, albo do jednego z jego synów.



Rys. 3

Podzielmy otrzymany ciąg  $a$  na fragmenciki o długości  $\frac{\log n}{2}$  każdy (sposób doboru tej wartości wyjaśni się dalej). Dla każdego fragmencika obliczymy minimum z elementów, które go tworzą i z tych minimów utworzymy nowy ciąg  $b$  o długości  $\frac{2n}{\log n}$ . Każdy segment  $S$  w  $a$  składa się z pewnej liczby całych fragmencików oraz maksymalnie dwóch kawałków fragmencików – możliwe sposoby takich rozkładów są przedstawione na rysunku 3. Odpowiadanie na zapytania RMQ dla ciągu  $a$  możemy zatem w dużej mierze sprowadzić do RMQ dla  $b$ , o ile założymy, że będziemy jakoś potrafili osobno zająć się maksymalnie dwoma niepełnymi fragmencikami na brzegach. Dla ciągu  $b$  możemy sobie pozwolić na wykonanie wstępnych obliczeń poprzednio otrzymanego algorytmu – złożoność czasowa tego kroku będzie równa  $O(m \log m)$ , gdzie  $m = \frac{2n}{\log n}$  to długość ciągu  $b$ . Wykonując podstawienie (i pomijając stały czynnik 2), otrzymujemy złożoność:

$$O\left(\frac{n}{\log n} \log\left(\frac{n}{\log n}\right)\right) = O\left(\frac{n}{\log n} \log n\right) = O(n).$$

W tym szacowaniu istotne było, że długość pojedynczego fragmencika ciągu jest logarytmicznego rzędu, co daje częściowe uzasadnienie takiego a nie innego wyboru jej wartości. W powyższym rozumowaniu pominęliśmy kwestię zamiany opisu segmentu  $S$  w ciągu  $a$  do segmentu  $S'$  w  $b$ , odpowiadającego pełnym fragmencikom składającym się na  $S$ ; nietrudno jednak zauważyć, że tę operację można wykonać w czasie stałym za pomocą kilku prostych wzorów (dokładny opis pozostawiamy Czytelnikowi).

Pozostał nam do rozpatrzenia sposób wyznaczania minimum z kawałków fragmencików, których maksymalnie dwa otrzymujemy przy każdym zapytaniu. Nie możemy tego kroku wykonywać siłowo, gdyż wówczas obsługa zapytań RMQ wymagałaby pesymistycznie wykonania  $\log n$  operacji, czyli nie miałyby żądanej złożoności czasowej  $O(1)$ . Zastanówmy się więc, ile jest *istotnie różnych* rodzajów fragmencików, jakie mogą powstać przy podziale naszego ciągu. Każdy taki fragmencik jest wyznaczony jednoznacznie przez swój element początkowy oraz przez ciąg złożony wyłącznie z jedynek i minus jedynek, będący ciągiem różnic między kolejnymi elementami fragmencika. Liczba różnych ciągów złożonych z 1 i  $-1$  o długości  $\frac{\log n}{2} - 1$  to:

$$2^{(\frac{\log n}{2} - 1)} = \frac{1}{2} \cdot (2^{\log n})^{1/2} = \frac{1}{2} \cdot n^{1/2} = \frac{1}{2} \sqrt{n}.$$

Każdy taki fragmencik możemy opisać jednoznacznie przez jedną liczbę całkowitą między 0 a  $\frac{1}{2} \sqrt{n} - 1$ , której cyfry w układzie dwójkowym determinują, czy odpowiednie różnice kolejnych elementów fragmencika są równe jeden, czy minus jeden. Dla każdego spójnego kawałka każdego typu fragmencika możemy teraz siłowo wyznaczyć minimum przy założeniu, że pierwszy element fragmencika jest równy 0; koszt czasowy wykonania takiego obliczenia to iloczyn liczby różnych możliwych fragmencików przez sześćcian z długości fragmencika, czyli  $O(\sqrt{n}(\log n)^3) = O(n)$ . Wszystkie otrzymane wartości możemy – dzięki prostej numeracji różnych fragmencików – spamiętać w trójwymiarowej tablicy (jej wymiary to typ fragmencika oraz początek i koniec jego kawałka), co da nam stały koszt odwołania się do pojedynczej jej komórki. To pokazuje także, skąd się wziął czynnik  $\frac{1}{2}$  w dobranej przez nas długości pojedynczego fragmencika: dzięki niemu złożoność czasowa wyznaczania opisanej tablicy pomocniczej nie jest większa niż liniowa.



**Rozwiązanie zadania M 1186.**

Przyjmijmy, że nieskończony ciąg  $a_n = x_n^2$  ( $n = 1, 2, \dots$ ), gdzie  $x_n > 0$ , spełnia warunki zadania. Liczba

$x_{n+1}^2 - x_n^2 = (x_{n+1} - x_n)(x_{n+1} + x_n)$  jest liczbą pierwszą lub kwadratem liczby pierwszej, a ponadto

$$0 < x_{n+1} - x_n < x_{n+1} + x_n.$$

Wobec tego  $x_{n+1} - x_n = 1$ , czyli

$$x_n = x_1 + n - 1.$$

Stąd obliczamy

$$x_{n+1}^2 - x_n^2 = x_{n+1} + x_n = 2x_1 + 2n - 1$$

dla  $n = 1, 2, \dots$ . Zatem gdyby ciąg  $(a_n)$  był nieskończony, to każda liczba nieparzysta większa lub równa  $2x_1 - 1$  byłaby liczbą pierwszą lub kwadratem liczby pierwszej. Uzyskana sprzeczność dowodzi, że rozpatrywany ciąg musi być skończony.



Na podstawie tak przeprowadzonych wstępnych obliczeń możemy już sobie poradzić z obsługą niepełnych fragmencików. Na początku dla każdego fragmencika w ciągu  $a$  wyznaczamy jego typ (liczbę całkowitą od 0 do  $\frac{1}{2}\sqrt{n} - 1$ ) oraz początkowy element. Przy obsłudze zapytania identyfikujemy jeden lub dwa niepełne fragmenciki, składające się na segment  $S$  z zapytania, następnie na podstawie ich typów i tego, jakie ich kawałki są zawarte w  $S$ , za pomocą pojedynczych odwołań do wyżej skonstruowanej tablicy wyznaczamy szukane minima. Ponieważ były one obliczone przy założeniu, że początkowy element fragmencika jest równy 0, to musimy je przeskalować o faktyczne początkowe elementy rozważanych fragmencików. Za pomocą kilku prostych wzorów możemy z opisu zapytania w czasie stałym wyłuskać potrzebne nam parametry dotyczące skrajnych fragmencików i ich kawałków, które są zawarte w segmencie, zatem cały ten krok może zostać wykonany dla każdego zapytania RMQ w złożoności czasowej  $O(1)$ . Łączny czas wszystkich wykonanych po drodze wstępnych obliczeń jest liniowy względem  $n$ , co pokazuje, że otrzymaliśmy algorytm, działający tak szybko, jak chcieliśmy. Jest to zarazem najszybszy algorytm na jaki można było praktycznie liczyć (nie sposób sobie wyobrazić istnienia algorytmu o asymptotycznie mniejszej niż liniowa złożoności czasowej wstępnych obliczeń), możemy zatem uznać, że znaleźliśmy najlepsze możliwe rozwiązanie zarówno problemu RMQ, jak i LCA.

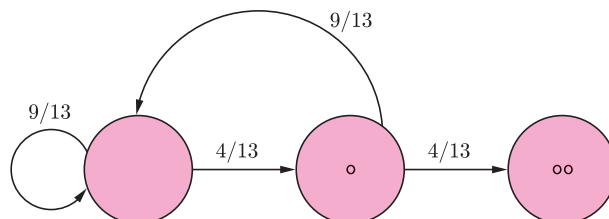
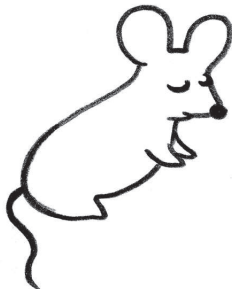
## Empiryczne badanie uogólnionej wersji gry Hamleta

*motywy*  
**Logo**

Andrzej WALAT

W październikowym numerze *Delty* rozważaliśmy następującą grę. Losujemy litery ze słynnej kwestii Hamleta: *to be or not to be* (być albo nie być) tak długo, aż otrzymamy dwuliterowe słowo *to*; za każde wylosowanie litery płacimy złotówkę, ale na końcu, po uzyskaniu słowa *to* otrzymujemy nagrodę  $n$  złotych. Ustaliliśmy, że aby gra była opłacalna, wartość nagrody  $n$  powinna być większa niż średnia liczba losowań, jakie trzeba wykonać, by otrzymać słowo *to*. Obliczyliśmy, że teoretyczna wartość tej średniej to  $\bar{x} = \frac{13}{3} \cdot \frac{13}{4} = \frac{169}{12} \approx 14,08$ .

Tym razem zajmiemy się innymi wariantami gry Hamleta. Na początek przyjmijmy, że naszym docelowym słowem jest *oo* i obliczmy średni czas oczekiwania na to słowo (tj. średnią liczbę losowań, jakie trzeba wykonać, by otrzymać *oo*). W tym przypadku gra Hamleta jest równoważna losowej wędrownicy pionka po planszy przedstawionej na rysunku 1 od pola początkowego  $x$  do pola końcowego  $oo$ ,



Rys. 1

a odpowiedni układ równań liniowych ma następującą postać:

$$\begin{cases} \bar{x} = 1 + \frac{4}{13}\bar{o} + \frac{9}{13}\bar{x} \\ \bar{o} = 1 + \frac{4}{13} \cdot 0 + \frac{9}{13}\bar{x} \end{cases}$$

Po jego rozwiązaniu otrzymujemy średni czas oczekiwania na dwuliterowe słowo oo:

$$\bar{x} = \frac{221}{16} = \frac{13}{4} + \left(\frac{13}{4}\right)^2.$$

Czy w grze Hamleta średni czas oczekiwania na trzyliterowe słowo ooo jest równy

$$\frac{13}{4} + \left(\frac{13}{4}\right)^2 + \left(\frac{13}{4}\right)^3?$$

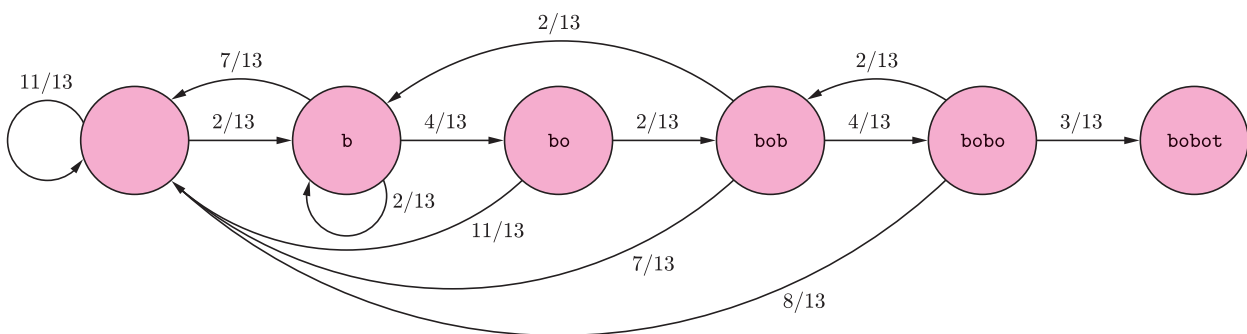


Fakt, że średni czas oczekiwania na dwuliterowe słowo oo okazał się krótszy niż odpowiedni średni czas oczekiwania na słowo to, chyba nikogo nie zdziwił. Te dwa słowa różnią się tylko pierwszą literą i litera o występuje w haśle *tobeornottobe* częściej niż litera t. Intuicja podpowiada nam, że na słowa utworzone z liter występujących w haśle z większą częstością będziemy średnio czekali krócej niż na słowa utworzone z liter występujących rzadziej. Czy wobec tego średni czas oczekiwania na ooo jest krótszy niż odpowiedni średni czas oczekiwania na słowo too?

Żeby odpowiedzieć na to pytanie, wystarczy ułożyć i rozwiązać dwa układy równań liniowych z trzema niewiadomymi. Ale potem pojawią się kolejne pytania i układanie oraz rozwiązywanie odpowiednich układów liniowych w końcu nam się znudzi. Dobrze byłoby mieć jakieś rozwiązanie ogólne, które dla dowolnego hasła oraz docelowego słowa daje średnią liczbę losowań, jakie trzeba wykonać, by wybierając losowo litery z hasła, utworzyć dane słowo docelowe. W tym artykule przedstawię empiryczne rozwiązanie tego problemu, a w kolejnym, grudniowym numerze *Delty* – rozwiązanie teoretyczne. W obu przypadkach posłużę się jako narzędziem językiem Logo.

## Rozwiązanie empiryczne

W ogólnym przypadku dane jest dowolne hasło i cel – słowo, które chcemy wygenerować, losując litery z hasła. Ten proces wymaga wielu losowań, nie mniej niż długość słowa docelowego, ale zwykle znacznie więcej. Wyróżniamy tyle różnych stanów procesu, ile jest różnych prefiksów danego słowa docelowego. Na przykład, jeśli dane jest hasło Hamleta *tobeornottobe* oraz cel *bobot*, to wyróżniamy 6 różnych stanów: słowo puste, jednoliterowe słowo *b* oraz *bo*, *bob*, *bobo*, *bobot*. Rysunek 2 przedstawia graf przejść pomiędzy stanami procesu.



Rys. 2

Naszym zadaniem jest zdefiniowanie funkcji *lilo*, która dla danego hasła i celu wyznacza losową liczbę kroków (losowań), po których zaczynając od stanu początkowego, osiągamy cel. Przedtem jednak rozwiążemy problem bardziej ogólny. Zdefiniujemy funkcję *lk*, która dla danego dowolnego stanu procesu, danego hasła oraz słowa docelowego wyznacza losową liczbę kroków od danego (aktualnego) stanu do celu. Wynik tej funkcji obliczamy w następujący sposób: sprawdzamy, czy dany (aktualny) stan jest docelowy, jeśli tak, to wynikiem funkcji jest liczba 0 i koniec, w przeciwnym przypadku trzeba wykonać jeden krok do jakiegoś wyznaczonego losowo następnego stanu – żeby obliczyć wynik, do liczby 1 musimy dodać liczbę kroków od następnego stanu do celu.



W Logo można to zakodować w następujący sposób.

```
oto lk :stan :hasło :cel
jeśli :stan = :cel [wynik 0]
wynik 1 + lk następny :stan :hasło :cel
już
```

A jak wyznaczyć następny stan procesu? Wybieramy losowo literę z hasła i dopisujemy ją na koniec aktualnego stanu. Wynikiem jest najdłuższy sufix otrzymanego w ten sposób słowa, który jest prefiksem danego słowa docelowego.



```
oto następny :stan
wynik suprefiks nak los :hasło :stan :cel
już
```

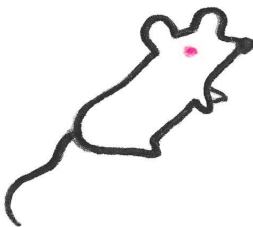
Brakuje jeszcze definicji funkcji suprefiks, która znajduje najdłuższy sufix pierwszego danego słowa będący prefiksem drugiego danego słowa.

```
oto suprefiks :s1 :s2
jeśli prefiks? :s1 :s2 [wynik :s1]
wy suprefiks bp :s1 :s2
już

oto prefiks? :s1 :s2
jeśli puste? :s1 [wy "prawda]
jeśli (pierw :s1) <> pierw :s2 [wy "fałsz]
wy prefiks? bp :s1 bp :s2
już
```

Obliczenie liczby kroków od stanu początkowego (czyli od słowa pustego) do celu jest tylko szczególnym przypadkiem zadania obliczenia liczby kroków od dowolnego stanu procesu do końca. Zapowiedzianą wyżej funkcję lilo można zdefiniować w następujący sposób.

```
oto lilo :hasło :cel
wynik lk " :hasło :cel
już
```



Możemy teraz eksperymentalnie sprawdzić, ile razy trzeba losować litery z hasła Hamleta, by wygenerować nasze przykładowe słowo bobot. Jednocześnie przetestujemy nasze rozwiązania programistyczne.

```
? powtórz 10 [wpisz lilo "tobeornottobe "bobot wpisz "|, |]
5678, 417, 1508, 727, 650, 2446, 1400, 1087, 2365, 2177,
```

Najszybciej otrzymaliśmy słowo bobot po 417 losowaniach, chociaż teoretycznie mogłoby się nam to udać już po pięciu. Największa liczba losowań to 5678. Trudno dokładnie ocenić odpowiednią średnią wartość liczby losowań na podstawie 10 wyników. Zdefiniujemy jeszcze jedną funkcję ślilo, której wynikiem jest średnia z  $n$  wartości funkcji lilo dla danego hasła i celu.

```
oto ślilo :n :hasło :cel
niech "suma 0
powtórz :n [przyp "suma :suma + lilo :hasło :cel]
wynik :suma / :n
już
```

Obliczmy średnią ze 100 wartości lilo. Po napisaniu polecenia:

```
pisz ślilo 100 "tobeornottobe "bobot
```

komputer wypisał 1798.37, ale po wywołaniu tego samego polecenia jeszcze raz komputer wypisał komunikat o błędzie:

Błąd w wierszu 2 procedury prefiks?: Zbyt wiele wywołań procedur. Prawdopodobnie nieskończona rekurencja.

Nasze procedury, chociaż są poprawne, mają ograniczony zakres stosowalności. Zadanie wyznaczania średniego czasu oczekiwania na słowo bobot leży



#### Rozwiązanie zadania F 704.

Początkowo tłoki znajdowały się w stanie równowagi, zatem były spełnione równania

$$2p_0 = p_0 + mg/S$$

oraz

$$3p_0 = p_0 + 2mg/S,$$

gdzie  $S$  jest polem przekroju poprzecznego naczynia. Po dociążeniu górnego tłoka, z prawa Boyle'a-Mariotte'a mamy, że:

$$3p_0 dS = l(Q + 2p_0 S),$$

$$2p_0 dS = (d - l)(Q + p_0 S),$$

gdzie  $l$  to nowa wysokość dolnego tłoka, a  $Q$  to siła dociskająca górny tłok.

Z powyższych równań otrzymujemy:

$$l = d(3 - \sqrt{6}) \approx 2,75 \text{ cm}.$$



na granicy tego zakresu. Można napisać inną iteracyjną wersję procedury `lilo`, by ten zakres znacznie poszerzyć. Pozostawiam to jako otwarte zadanie dla Ambitnego Czytelnika. Tymczasem posłużymy się funkcją `ślilo`, by odpowiedzieć na kilka pytań dotyczących gry Hamleta (i nie tylko).



### 1. Na które słowo: `too` czy `ooo` trzeba średnio dłużej czekać?

Żeby to sprawdzić, napiszmy:

```
pisz ślilo 1000 "tobeornottobe "too
```

```
44.858
```

a następnie:

```
pisz ślilo 1000 "tobeornottobe "ooo
```

```
48.201
```



Wynik tego pojedynku jest korzystny dla `too`. Średni czas oczekiwania na `too` w tysiącu próbach okazał się krótszy niż w przypadku `ooo`. Wątpiącemu Czytelnikowi, który ma trudność z pogodzeniem tego faktu z intuicją, proponuję narysowanie dwóch grafów podobnych, jak na rysunkach 1 i 2, reprezentujących procesy losowania liter z hasła Hamleta, aż do uzyskania odpowiedniego słowa `too` oraz `ooo`, a następnie wyobrażenie sobie, jak mogą przebiegać te dwa procesy.



### 2. Czy średni czas oczekiwania na słowa: `too`, `oto` oraz `oot` jest taki sam?

Tym razem napiszemy po kolei następujące dwa polecenia:

```
pisz ślilo 1000 "tobeornottobe "oot
```

```
44.709
```

```
pisz ślilo 1000 "tobeornottobe "oto
```

```
49.081
```



Średni czas oczekiwania na słowo `oot` okazał się zbliżony do otrzymanego przed chwilą średniego czasu oczekiwania na `too`. Średni czas oczekiwania na `oto` okazał się wyraźnie dłuższy. Czy te eksperymentalne wyniki dają podstawę do twierdzenia, że odpowiednie teoretyczne średnie czasy oczekiwania na `too` oraz `oot` są równe i mniejsze niż teoretyczny średni czas oczekiwania na `oto`? Tym i innymi podobnymi zagadnieniami zajmiemy się w następnym numerze. Ambitny Czytelnik może potraktować je jako zadanie domowe.



Na koniec zajmiemy się jeszcze jednym problemem, który z pozoru nie ma związku z grą Hamleta.



### 3. Ile razy średnio trzeba rzucić kostką, żeby otrzymać trzy szóstki po kolei?

Teoretycznie:  $6 + 6^2 + 6^3 = 258$ .

Posłużymy się funkcją `ślilo`, by skonfrontować ten teoretyczny wynik z doświadczeniem. W przypadku, gdy hasłem jest sześcioliterowe słowo utworzone z kolejnych cyfr `123456`, a słowo docelowe to sekwencja trzech szóstek `666`, gra Hamleta jest równoważna z rzucaniem kostką aż do uzyskania trzech szóstek po kolei. Użyjemy funkcji `ślilo`, by obliczyć 10 razy średni czas oczekiwania na trzy szóstki w serii 1000 prób.



```
powtórz 10 [wpisz ślilo 1000 123456 666 wpisz "|, |]
```

```
253.904, 269.942, 257.992, 256.343, 263.492, 258.231, 252.877, 257.92, 263.571, 260.772,
```



Otrzymane empiryczne średnie są dość zgodne ze średnią teoretyczną, ale jak zwykle tylko do pewnego stopnia. Największa otrzymana eksperymentalnie średnia różni się od dokładnej wartości teoretycznej o około 5%. Rzucanie kostką aż do uzyskania określonej sekwencji wyników nie jest jedynym przykładem procesu losowego, który sprowadza się do gry Hamleta. To jeden z powodów, by się nią zainteresować. W kolejnym odcinku wrócimy do tej gry i będziemy ją rozważali z perspektywy teoretycznej.



## Informatyczny kącik olimpijski (2)

Oto zadanie z Bałtyckiej Olimpiady Informatycznej (2004):



Mamy dany ciąg liczb naturalnych  $a_i$ ,  $1 \leq i \leq N$ . Należy znaleźć taki ciąg  $b_i$ , żeby wartość

$$\sum_{i=1}^N |a_i - b_i|$$

była najmniejsza możliwa oraz  $b_i \leq b_{i+1}$  dla  $1 \leq i < N$ .

Na początek zajmijmy się trochę prostszym zadaniem. Oznaczmy  $S = \sum_{i=1}^N |a_i - b_i|$  i spróbujmy obliczyć minimalną możliwą wartość  $S$ . Zadanie to ma proste rozwiązanie, korzystające z metod programowania dynamicznego. Niech liczba  $A_k(v)$  oznacza minimalną wartość sumy  $\sum_{i=1}^k |a_i - b_i|$ , przy założeniu, że  $b_k = v$ , dla  $1 \leq k \leq N$ . Dla ułatwienia obliczeń przyjmijmy, że  $A_0(v) = 0$  dla dowolnego  $v$ . Łatwo zauważyć, że optymalny ciąg  $b_i$  nie ma wartości mniejszych, niż najmniejsze  $a_i$ , ani większych, niż największe  $a_i$ . Oznaczmy te najmniejsze i największe  $a_i$ , odpowiednio, przez  $X$  i  $Y$ , zaś  $T$  niech oznacza liczbę możliwych wartości, jakie mogą przybierać elementy ciągu  $a_i$  ( $T = Y - X + 1$ ). Zauważmy też, że optymalny ciąg  $b_i$  nie zawiera żadnych wartości poza wartościami występującymi w  $a_i$  – ta obserwacja przyda nam się później. Teraz  $S = \min_{X \leq v \leq Y} A_N(v)$ , zaś na  $A_k(v)$ , dla  $1 \leq k$ , mamy prosty wzór:

$$A_k(v) = |v - a_k| + \min_{X \leq v' \leq v} A_{k-1}(v').$$

Wzór ten daje nam algorytm, który w czasie  $O(NT)$  oblicza minimalne możliwe  $S$  (jeśli tylko, dla  $v > X$ , będziemy pamiętać minimalne  $A_{k-1}(v')$ , co pozwoli obliczyć  $A_k(v)$  na podstawie  $A_k(v-1)$  w czasie stałym). Potrzebujemy tylko  $O(T)$  pamięci, ponieważ do wyliczenia wartości  $A_k$  potrzebne są tylko wartości  $A_{k-1}$ , które możemy nadpisać przy zwiększaniu  $k$ . Jeśli jednak chcemy znaleźć sam ciąg  $b_i$ , musimy dla każdego  $A_k(v)$  zapamiętać, jakie  $v'$  daje taką jego wartość. Liczby  $v$  i  $v'$  to nic innego, jak wartości  $b_k$  i  $b_{k-1}$ . W ten sposób złożoność pamięciowa rośnie do  $O(NT)$ , a czasowa pozostaje taka sama.

Dodatkowo, możemy obniżyć złożoność, ograniczając  $T$ . Jak zauważyliśmy, wyrazy ciągu  $b_i$  nie przyjmują innych wartości, jak te występujące w ciągu  $a_i$ . Tych wartości jest co najwyżej  $N$ . Możemy więc obliczać wartości  $A_k(v)$  tylko dla  $O(N)$  różnych wartości  $v$ . W ten sposób, po wstępnym „wyjęciu” i posortowaniu możliwych wartości elementów ciągu  $b_i$  w czasie  $O(N \log N)$ , główna część algorytmu będzie mieć złożoność czasową i pamięciową  $O(N^2)$ .

Ale istnieje też rozwiązanie o lepszej złożoności. Na początek dziedzinę  $A_k$  rozszerzmy na wszystkie liczby naturalne, nie tylko te, które leżą w przedziale  $[X, Y]$ . Wzór  $A_k(v) = |v - a_k| + \min_{v' \leq v} A_{k-1}(v')$  wciąż ma sens, ponieważ dla  $v < X$  liczba  $A_k(v)$  jest coraz większa dla coraz mniejszych  $v$  (poza przypadkiem, gdy

$k = 0$ ). Oznaczmy  $f \diamond(v) = f(v-1) + f(v+1) - 2f(v)$ . Jest to podwójna pochodna dyskretna ciągu  $f$ , przesunięta o 1 pozycję. Ta „pochodna” ma własność:

$$f \diamond + g \diamond = (f + g) \diamond.$$

co widać wprost z definicji  $f \diamond$ . Zastanówmy się, jak – łatwo i szybko – znając  $A_k \diamond$ , znaleźć  $A_{k+1} \diamond$ ? Przejście od  $A_k$  do  $A_{k+1}$  podzielimy na dwa etapy – najpierw wyliczymy  $B_k(v) = \min_{v' \leq v} A_k(v')$ , a następnie do tego ciągu dodamy ciąg  $C_k(v) = |v - a_{k+1}|$ , otrzymując  $A_{k+1}$ . Stąd  $C_k \diamond$  jest ciągiem samych zer, poza jedynym wyrazem  $C_k \diamond(a_{k+1}) = 2$ . Ciąg  $B_0$  składa się z samych zer, a więc  $A_1 = C_0$ . Zastanówmy się teraz, jak wyglądają ciągi  $A_k$  dla  $k > 0$ . Dla odpowiednio dużych  $v$  mamy  $A_k(v+1) - A_k(v) = 1$ , ponieważ ciąg  $b_i$ , dla  $1 \leq i < k$  zachowuje optimum, i zmienia się tylko  $b_k$ . Co więcej, jak się za chwilę okaże, ciąg  $A_k \diamond$  ma wyrazy nieujemne. Niech  $t_k$  będzie największym indeksem takim, że  $A_k \diamond(t_k) > 0$ . W takiej sytuacji  $B_k(v) = A_k(v)$  dla każdego  $v \leq t_k$ , oraz  $B_k(v) = A_k(t_k)$  dla pozostałych  $v$ . Podobnie, dla  $v \neq t_k$  mamy  $A_k \diamond(v) = B_k \diamond(v)$ , a dla  $v = t_k$  mamy  $A_k \diamond(v) = B_k \diamond(v) + 1$ . A skoro  $A_{k+1} \diamond = B_k \diamond + C_k \diamond$ , to  $A_{k+1} \diamond$  ma wyrazy nieujemne. Warto zauważyć w tym momencie, że  $A_k(t_k)$  jest najmniejszą wartością ciągu  $A_k$ .

Jak teraz opisać funkcję  $A_k \diamond$ ? Zapamiętajmy ją jako multizbiór – jeśli funkcja ma na pozycji 4 wartość 2, to w zbiorze znajdują się dwie czwórki itp. Przejście z  $A_k \diamond$  do  $B_k \diamond$  wymaga znalezienia największego elementu i usunięcia go (tylko raz, jeśli się powtarza). Przejście z  $B_k \diamond$  do  $A_{k+1} \diamond$  to dodanie do zbioru dwóch wartości  $a_{k+1}$ . W takim razie widać już, jak znaleźć opis funkcji  $A_N \diamond$  w czasie  $O(N \log N)$  i pamięci  $O(N)$ . Ile wynosi zatem najmniejsze możliwe  $S$ ? Oczywiście, jest to  $A_N(t_N)$ . Jak z kolei wyliczyć kolejne wartości  $A_k(t_k)$ ? Wiemy na pewno, że

$$A_{k+1}(t_{k+1}) = B_k(t_{k+1}) + |t_{k+1} - a_{k+1}|.$$

Jeśli jeszcze zauważymy, że dla  $k < N$  mamy  $A_k(t_k) = B_k(t_k) = B_k(t_{k+1})$ , to nie tracąc na złożoności możemy rozszerzyć algorytm obliczania  $A_N \diamond$  o obliczanie  $A_k(t_k)$ . Podobnie, nie tracąc nic na złożoności, możemy skorzystać ze spostrzeżenia, że  $b_N = t_N$ , a dla  $k < N$  mamy  $b_k = \min(t_k, b_{k+1})$ , i obliczyć również optymalny ciąg  $b_k$ .

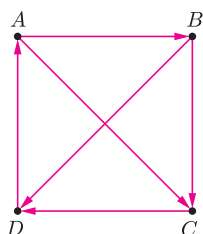
Filip WOLSKI



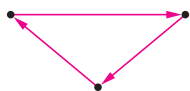
# mała delta

## Ilu liderów?

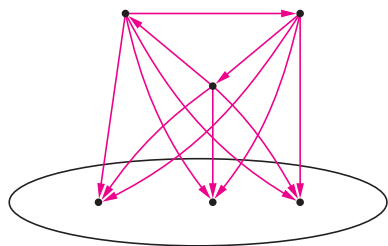
W siatkarskich rozgrywkach ligowych lub w innym turnieju, w którym każdy gra z każdym, startuje  $n$  drużyn. W siatkówce, co ważne, nie ma remisów. Zwykle nie ma zespołu, który rozgromiłby wszystkie inne, stąd łagodniejsza definicja.



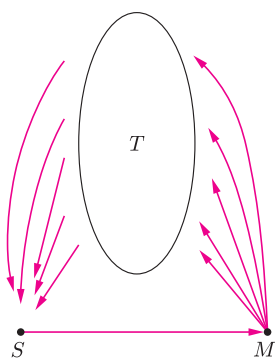
Rys. 1



Rys. 2



Rys. 3



Rys. 4

**Definicja.** Drużynę  $A$  nazwiemy *liderem*, jeśli każdą inną drużynę  $B$  pokonała ona bądź w bezpośrednim pojedynku, bądź istnieje taka drużyna  $C$ , że  $A$  wygrała z  $C$ , a  $C$  z  $B$ .

Wyniki rozgrywek obrazuje pełny graf skierowany (rys. 1), w którym wierzchołkom przypisujemy drużyny, a krawędzie opisują wyniki meczów (strzałka prowadzi od zwycięzcy do pokonanego). Taki graf nazywamy fachowo właśnie *turniejem*. W turnieju z rysunku 1 drużyna  $A$  jest liderem, bo pokonała bezpośrednio  $B$  i  $C$ , pośrednio zaś  $D$ . Liderami są też drużyny  $B$  i  $D$ , a nie jest nim zespół  $C$ .

Ilu maksymalnie może być liderów w turnieju z  $n$  drużynami? Trzy drużyny mogą wygrywać ze sobą „w kółku” (rys. 2), a idąc dalej tym tropem, dla każdego  $n \geq 3$  można skonstruować turniej, w którym liderów będzie trzech (rys. 3). Czy to już maksimum? Otóż nie.

**Fakt.** Jeśli  $n \neq 2$  i  $n \neq 4$ , to istnieje taki  $n$ -drużynowy turniej, w którym wszystkie (!) zespoły są liderami.

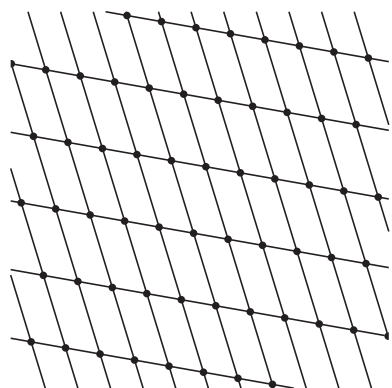
Oto dowód. Do  $n$ -drużynowego turnieju  $T$ , w którym wszyscy są liderami, dołączmy dwie nowe drużyny, Mistrzów i Słabeuszy (rys. 4). Wyniki spotkań ustawiamy (byle tylko nie było z tego jakiejś afery) następująco: Mistrzowie pokonują wszystkie drużyny z turnieju  $T$ , Słabeusze zaś z nimi wszystkimi przegrywają. W pojedynku Mistrzów ze Słabeuszami wygrywają... Słabeusze (to podejrzane). Łatwo zobaczyć, że w nowym turnieju każda z drużyn jest liderem.

Dowód faktu zakończy wskazanie turnieju z trzema (rys. 2) i sześcioma (zadanie domowe) drużynami, w którym każdy zespół jest liderem. Obrazek będzie pełniejszy, jeśli Czytelnik udowodni ponadto, że dla  $n = 2$  lub  $n = 4$  nie ma takich turniejów (wymaga to w każdym z tych przypadków  $n - 2$  minut zastanowienia).

Choć wszyscy są liderami, to wyniki naszego turnieju nie są wcale wyrównane: Szczyt tabeli ligowej okupują Mistrzowie, z  $n - 2$  wygranymi w  $n - 1$  meczach, a w ogonie ciągnie się drużyna Słabeuszy, z tylko jedną wygraną. Czytelnik zechce skonstruować taki  $n$ -drużynowy turniej, w którym każda drużyna będzie liderem, a do tego tabela wyników będzie możliwie wyrównana (dla nieparzystego  $n$  może być nawet całkiem płaska – wszyscy z tą samą liczbą punktów).

## O pożytkach z siatki

Jednakowymi równoległobokami łatwo pokryć płaszczyznę (rys. 1). Pokrycie takie nazywamy *siatką*, a wszystkie wierzchołki równoległoboków nazywamy jej *wierzchołkami*. Jeśli pole równoległoboku jest równe 1, to wyznaczoną przez niego siatkę nazywamy *jednostkową*. W układzie współrzędnych najbardziej rzuca się w oczy siatka jednostkowa złożona z kwadratów, których wierzchołkami są wszystkie punkty o obu współrzędnych całkowitych. Mniej oczywiste, ale mimo to prawdziwe jest następujące twierdzenie.



Rys. 1

**Twierdzenie o siatkach jednostkowych.** *Wszystkie wierzchołki dowolnej siatki jednostkowej, której jednym z wierzchołków jest początek układu współrzędnych, mają współrzędne*

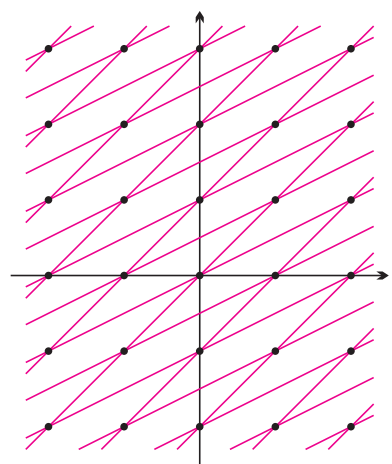
$$(a \cdot m + b \cdot n, c \cdot m + d \cdot n),$$

gdzie  $a, b, c$  i  $d$  są dowolnie ustalonymi liczbami spełniającymi warunek

$$(*) \quad |a \cdot d - b \cdot c| = 1,$$

liczby zaś  $m$  i  $n$  przebiegają wszystkie liczby całkowite.

Oczywiście, „zwykłą” siatkę jednostkową otrzymamy dla  $a = d = 1$  i  $b = c = 0$ . Na rysunku 2 widzimy też siatkę otrzymaną dla  $a = b = d = 1$  i  $c = 2$  (rzeczywiście  $|1 \cdot 1 - 1 \cdot 2| = 1$ ). Liczby  $a, b, c, d$  nie muszą jednak być całkowite ani nawet wymierne – byle tylko spełniały warunek (\*). To spostrzeżenie pozwala na wykorzystanie siatek do stwierdzenia, jak dalece można liczbę niewymierną przybliżyć przez liczbę wymierną – za chwilę to sprecyzujemy.

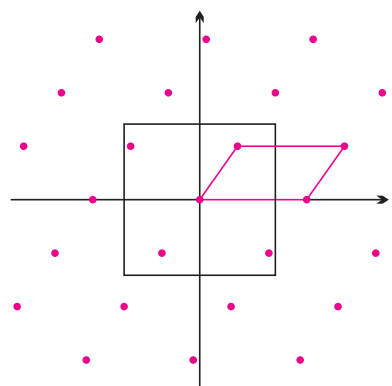


Rys. 2

Potrzebne do tego jest jeszcze jedno twierdzenie.

**Twierdzenie Minkowskiego o kwadracie.** *Kwadrat o boku 2, mający środek w wierzchołku dowolnie obranej siatki jednostkowej, ma w swoim wnętrzu lub na brzegu jeszcze co najmniej jeden wierzchołek tej siatki.*

Twierdzenie to wydaje się oczywiste, ale – o dziwo – o ile dowód poprzedniego twierdzenia jest dość banalny (proszę spróbować – trzeba zwyczajnie obliczyć pole równoległoboku), o tyle dowód twierdzenia Minkowskiego, ze względu na dowolność siatki, nie jest całkiem prosty. Ale my oba twierdzenia przyjmiemy bez dowodu (dowód ogólniejszej wersji drugiego twierdzenia znajduje się na stronie obok).



Rys. 3. Współrzędne wierzchołków kolorowego równoległoboku to  $(0, 0)$ ,  $(\frac{\alpha}{\varepsilon}, \varepsilon)$ ,  $(\frac{\alpha+1}{\varepsilon}, \varepsilon)$ ,  $(\frac{1}{\varepsilon}, 0)$ .

Weźmy teraz jakąś liczbę niewymierną  $\alpha$ . Ponadto obierzmy sobie jakąś (naprawdę dowolną!) dodatnią liczbę  $\varepsilon$  i zbudujmy siatkę jednostkową (rys. 3) dla

$$a = \frac{-1}{\varepsilon}, \quad b = \frac{\alpha}{\varepsilon}, \quad c = 0, \quad d = \varepsilon.$$

Istotnie, siatka jest jednostkowa, bo  $ad - bc = ad = -1$ . Zgodnie z twierdzeniem Minkowskiego, wewnątrz lub na brzegu kwadratu o wierzchołkach  $(1, 1)$ ,  $(-1, 1)$ ,  $(-1, -1)$ ,  $(1, -1)$  znajduje się jakiś punkt z naszej siatki – oznaczmy odpowiadające mu liczby całkowite przez  $m_0$  i  $n_0$ . Obie jego współrzędne muszą mieścić się w przedziale  $\langle -1; 1 \rangle$ . Mamy zatem

$$\left| \frac{-m_0}{\varepsilon} + \frac{\alpha \cdot n_0}{\varepsilon} \right| \leq 1 \quad \text{i} \quad |\varepsilon \cdot n_0| \leq 1,$$

czyli ( $\varepsilon$  jest dodatnie!)

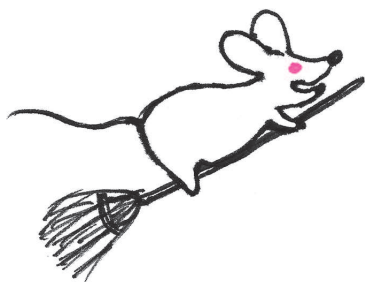
$$(**) \quad |\alpha \cdot n_0 - m_0| \leq \varepsilon \leq \frac{1}{|n_0|},$$

skąd (dzieląc przez  $n_0$ ) otrzymujemy

$$\left| \alpha - \frac{m_0}{n_0} \right| \leq \frac{1}{n_0^2}.$$

Okazało się zatem, że dla dowolnej liczby niewymiernej  $\alpha$  istnieje taka liczba wymierna  $\frac{m_0}{n_0}$ , która przybliży tę liczbę niewymierną z dokładnością co najmniej taką, jak odwrotność kwadratu swojego mianownika.

Warto zwrócić uwagę na rolę dowolnie obieranego  $\varepsilon$ . Jeśli będzie ono coraz mniejsze, to – wobec (\*\*\*) – otrzymywane przybliżenia wymierne będą coraz bliższe  $\alpha$ .



## Czarownica w nieskończonym lesie

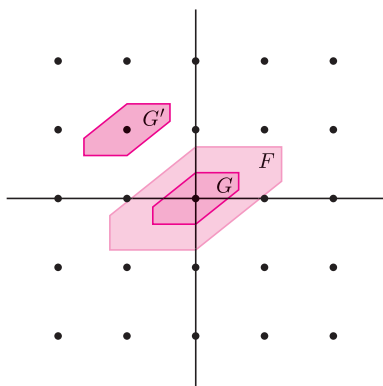
Marcin PILIPCZUK\*

Dawno, dawno temu, kiedy Ziemia była jeszcze płaszczyzną, rósł nieskończony las: w każdym punkcie kratowym Ziemi (czyli punkcie o obu współrzędnych całkowitych) rosło drzewo.

Czarownica Mirosława chciała zbudować sobie dom na drzewie. Czarownica ma jednak bardzo dużo wymagań co do nowego domu:

- ma być wielokątem wypukłym, by nie dało się w nim zgubić;
- ma być zbudowany na jednym drzewie, żadne inne drzewo nie ma pnia wewnątrz budynku, bo to głupio wygląda;
- dodatkowo, by dom się nie przewrócił, powinien on być środkowosymetryczny względem pnia drzewa, na którym się opiera;
- jego powierzchnia powinna być jak największa, w końcu Czarownice lubią żyć wygodnie.

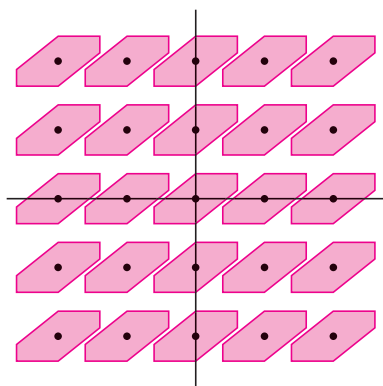
Czy Czarownicy uda się zbudować dom marzeń? Sformułujmy zadanie w języku matematyki: szukamy jak największego wielokąta wypukłego środkowosymetrycznego względem punktu  $(0, 0)$ , niezawierającego innego punktu kratowego.



Weźmy taki wielokąt  $F$ . Oznaczmy  $G = \frac{1}{2}F$ , czyli  $G$  to  $F$  zmniejszony dwukrotnie przez jednokładność względem punktu  $(0, 0)$ . Dla  $x_0, y_0 \in \mathbb{Z}$  oznaczmy  $G' = G + (x_0, y_0)$ , czyli  $G'$  to  $G$  przesunięty o wektor  $(x_0, y_0)$ ; innymi słowy, przesuwamy wielokąt  $G$  tak, by jego środek był w punkcie  $(x_0, y_0)$ . Załóżmy, że  $(x, y) \in G \cap G'$ . Wówczas  $A = (2x, 2y) \in F$  i  $B = (2(x - x_0), 2(y - y_0)) \in F$ . Wielokąt  $F$  był środkowosymetryczny, czyli  $A' = (-2x, -2y) \in F$ . Z warunku wypukłości  $F$  wynika, że środek odcinka  $A'B$  należy do  $F$ . Środek ten ma współrzędne:

$$\left( \frac{2(x - x_0) - 2x}{2}, \frac{2(y - y_0) - 2y}{2} \right) = (-x_0, -y_0).$$

Jest to punkt kratowy! Czyli, skoro  $F$  było poprawnym domkiem Czarownicy, to  $x_0 = y_0 = 0$ . Udowodniliśmy, że równoległe, przesunięte o wektor całkowity, kopie wielokąta  $G$  są rozłączne. Intuicja już nam podpowiada, że wobec tego domek Czarownicy nie może być za duży. Spróbujmy to pokazać.



Domek jest ograniczony – bo jest wielokątem; załóżmy, że zawiera się w kwadracie  $[-m, m] \times [-m, m]$ . Weźmy bardzo, bardzo duże  $M$ , dużo większe od  $m$ . Dla każdego punktu kratowego  $(x_0, y_0)$ , spełniającego  $-M \leq x_0, y_0 \leq M$ , tworzymy kopię wielokąta  $G$  o środku w tym punkcie kratowym. Kopie te są rozłączne – to udowodniliśmy – i zawierają się w kwadracie  $[-M - m, M + m]^2$ , o polu  $4(M + m)^2$ . Kopii stworzyliśmy  $(2M + 1)^2 > 4M^2$ . Kopie mają równe pola; czyli kopia nie może mieć większego pola niż

$$\frac{4(M + m)^2}{4M^2} = \left(1 + \frac{m}{M}\right)^2,$$

co dla dużego  $M$  jest dowolnie bliskie jedności. Czyli  $G$  nie może mieć większego pola niż 1.  $G$  było pomniejszonym dwukrotnie wielokątem  $F$ , więc  $F$  ma maksymalnie pole 4. Biedna Czarownica.

Zostawiając Czarownicę Mirosławę z jej problemem, spróbujmy uogólnić nasz wynik. Po pierwsze zauważmy, że nigdzie nie korzystaliśmy z tego, że  $F$  jest wielokątem, była dla nas istotna wypukłość, środkowosymetryczność i ograniczoność. Po drugie, spróbujmy przeprowadzić nasz dowód w  $n$  wymiarach. Niewiele się zmienia: będziemy mieli  $(2M + 1)^n$  kopii zbioru  $G$  w kostce o objętości  $2^n(M + m)^n$ ; więc wciąż objętość  $G$  nie może przekraczać jedności, czyli objętość domku Czarownicy – zbioru  $F$  – wynosi maksymalnie  $2^n$ .

\*student, Wydział Matematyki, Informatyki i Mechaniki, Uniwersytet Warszawski



### Rozwiązanie zadania M 1188.

Ustalmy liczbę całkowitą dodatnią  $k$  oraz rozpatrzmy zbiory  $A_1 = \{1, k+1\}$ ,  $A_2 = \{2, k+2\}, \dots, A_k = \{k, 2k\}$ . Liczby  $x_1 < x_2 < \dots < x_{k+1}$  są dodatnie, jest ich  $k+1$  oraz żadna z nich nie przekracza  $2k$ . Wobec tego, na mocy zasady szufladkowej, pewne dwie liczby  $x_i < x_j$  znajdują się w tym samym zbiorze  $A_l$ . To oznacza, że  $x_j - x_i = k$ , co kończy dowód.

## Euler, Lagrange i trzy ciała

Mikołaj KORZYŃSKI\*

W *Delcie* 10/2007 Tomasz Kwast napisał o problemie trzech ciał, a właściwie jego szczególnej wersji: o tzw. zredukowanym problemie trzech ciał.

Rozwiązania, które opisał, można stosunkowo łatwo uogólnić i to na dwa sposoby: rezygnując z założenia o znikomej masie trzeciego ciała i dopuszczając szerszą klasę orbit niż kołowe. Ze względu na ich urodę postaram się te uogólnione rozwiązania opisać w tym artykule. Najpierw jednak trochę historii.

Problem trzech ciał jest dla mechaniki tym, czym wielkie twierdzenie Fermata dla algebry i teorii liczb: postawionym dawno, pozornie prostym zagadnieniem, którego badanie przez niemal dwieście lat wpływało na rozwój nowych technik matematycznych, i za które brali się najwybitniejsi: Poincaré, Jacobi, Weierstrass, a wcześniej wspomniani w tytule Euler i Lagrange. Oczywiście, pełnego rozwiązania problemu nie ma i nie będzie, gdyż ruch jest w ogólności chaotyczny. Dwaj najwięksi mechanicy XVIII wieku znaleźli jednak pierwsze, ścisłe rozwiązania szczególne.

Za nimi założymy, że ruch wszystkich ciał odbywa się w jednej płaszczyźnie. Zapytamy, czy jest możliwe, aby ruch był wyjątkowo prosty: aby ciała tworzyły cały czas pewną figurę geometryczną, poddawaną zmiennym w czasie obrotom i jednokładnościom. Pojedynczo każde ciało wykonywać będzie ruch wokół środka masy, przy czym przyspieszenie skierowane będzie w kierunku tego środka.

Oznaczmy wektory położenia trzech ciał przez  $\vec{x}_i$ , ich masy przez  $m_i$ , a  $r_{jk} = |\vec{x}_j - \vec{x}_k|$  niech oznacza odległość  $j$ -tego ciała od  $k$ -tego. Równania ruchu mają postać

$$\begin{aligned}\ddot{\vec{x}}_1 &= \frac{Gm_2}{r_{12}^3} (\vec{x}_2 - \vec{x}_1) + \frac{Gm_3}{r_{13}^3} (\vec{x}_3 - \vec{x}_1), \\ \ddot{\vec{x}}_2 &= \frac{Gm_1}{r_{12}^3} (\vec{x}_1 - \vec{x}_2) + \frac{Gm_3}{r_{23}^3} (\vec{x}_3 - \vec{x}_2), \\ \ddot{\vec{x}}_3 &= \frac{Gm_1}{r_{13}^3} (\vec{x}_1 - \vec{x}_3) + \frac{Gm_2}{r_{23}^3} (\vec{x}_2 - \vec{x}_3).\end{aligned}$$

Zgodnie z zasadą zachowania pędu w pewnym układzie inercjalnym środek masy trzech mas spoczywa. Wybierzmy ten układ i dla uproszczenia przyjmijmy, że środek masy pokrywa się ze środkiem układu współrzędnych, czyli

$$(1) \quad m_1 \vec{x}_1 + m_2 \vec{x}_2 + m_3 \vec{x}_3 = 0.$$

Aby trzy masy w dowolnych dwóch momentach tworzyły figury podobne (np. trójkąty) o ustalonym środku masy, potrzebne jest, aby ich prędkości i przyspieszenia również były tak samo przekształcone. W szczególności oznacza to, że wektory ich przyspieszenia są proporcjonalne do położenia z tym samym współczynnikiem proporcjonalności dla wszystkich ciał, czyli

$$(2) \quad \ddot{\vec{x}}_k = -a \vec{x}_k$$

dla pewnej stałej  $a$ . Porównanie tej zależności z równaniami ruchu daje

$$\begin{aligned}\frac{Gm_2}{r_{12}^3} \vec{x}_2 + \frac{Gm_3}{r_{13}^3} \vec{x}_3 &= \left(-a + \frac{Gm_2}{r_{12}^3} + \frac{Gm_3}{r_{13}^3}\right) \vec{x}_1 \\ m_2 \vec{x}_2 + m_3 \vec{x}_3 &= -m_1 \vec{x}_1\end{aligned}$$

i podobnie dla pozostałych par ciał. Dla porównania dopisaliśmy na dole przekształcone równanie środka masy (1).

Rozważmy dwa przypadki. Jeśli wszystkie odległości są równe ( $r_{12} = r_{23} = r_{13}$ ), to pierwsze z tych równań wynika prosto z drugiego przez pomnożenie stronami przez stałą. Drugie równanie jest zawsze prawdziwe – środek masy wszak się nie przesuwa – więc i pierwsze jest spełnione tożsamościowo. Postawione warunki spełnia więc konfiguracja z masami umieszczonymi w wierzchołkach trójkąta równobocznego.

\*Wydział Fizyki, Uniwersytet Warszawski



Jeśli jednak nie wszystkie odległości są równe, powiedzmy  $r_{12} \neq r_{13}$ , to równania są niezależne i metodą eliminacji można wyznaczyć  $\vec{x}_2$  i  $\vec{x}_3$ . Konkretnie wzory nie są tu istotne, ważne, że oba wektory muszą być proporcjonalne do  $\vec{x}_1$ . Innymi słowy, jeśli ciała nie tworzą trójkąta równobocznego, to muszą znajdować się na jednej prostej. Dopuszczalne wzajemne odległości można dalej wyznaczyć, korzystając z powrotu z (2), ale nie będziemy tutaj tego robić, pozostawiając to Czytelnikowi.

Od nazwisk odkrywców konfiguracje pierwszego typu przyjęło się nazywać lagranżowskimi, drugiego eulerowskimi.

Jak w takim przypadku wygląda trajektoria pojedynczego ciała? Dla prostoty rozważymy tylko przypadek „trójkątny”, w konfiguracji eulerowskiej wszystko wygląda całkiem podobnie, choć rachunki są nieco bardziej skomplikowane.

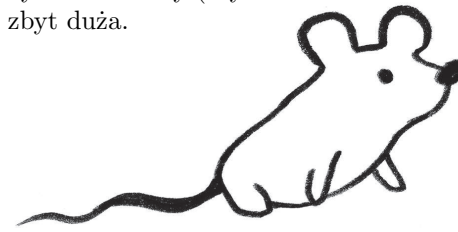
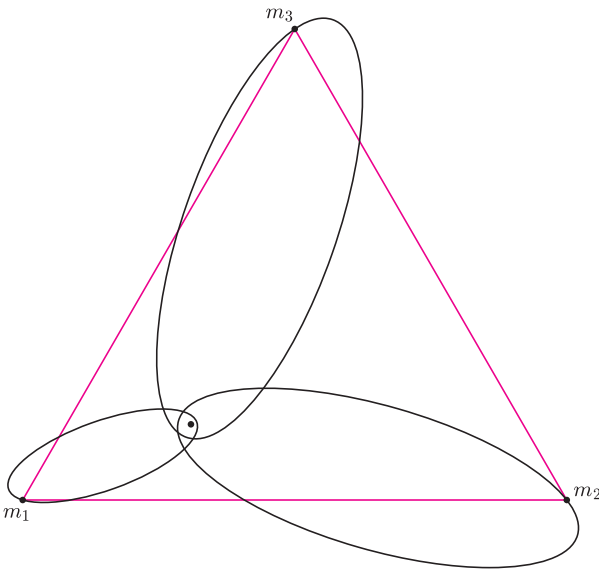
Niech  $r$  oznacza bok trójkąta. Równanie ruchu ciała 1 przyjmuje postać

$$\begin{aligned} \ddot{\vec{x}}_1 &= \frac{Gm_2}{r^3} (\vec{x}_2 - \vec{x}_1) + \frac{Gm_3}{r^3} (\vec{x}_3 - \vec{x}_1) = -\frac{G(m_1 + m_2 + m_3)}{r^3} \vec{x}_1 = \\ &= -\frac{|\vec{x}_1|^3}{r^3} G(m_1 + m_2 + m_3) \frac{\vec{x}_1}{|\vec{x}_1|^3}. \end{aligned}$$

W drugim przekształceniu skorzystaliśmy jeszcze raz z (1). W rozważanych przez nas rozwiązaniach w czasie ruchu trójkąt obraca się i przekształca przez jednokładność względem środka układu współrzędnych. Iloraz  $\frac{|\vec{x}_1|^3}{r^3}$  nie zmienia się pod wpływem tych przekształceń, pozostaje więc cały czas stały. Ostatnia postać równania pokazuje więc, że siła działająca na masę 1, a więc i pozostałe, jest taka, jakby w środku układu współrzędnych znajdowało się jedno ciężkie, nieruchome ciało.

Czytelnicy zaznajomieni z problemem Keplera wiedzą, że w takiej sytuacji ruch ciał będzie odbywać się po krzywej stożkowej o ognisku w środku układu współrzędnych. Oczywiście, wszystkie trzy ciała zakreślają jednocześnie trzy stożkowe o tym samym mimośrodku. W szczególności możliwy jest ruch po okręgu opisany w artykule Tomasza Kwasta.

Pozostaje pytanie o stabilność takiego ruchu, czyli czy drobne zaburzenie nie spowoduje rozpadu tak ładnej konfiguracji. Okazuje się, że jedynie lagranżowskie ruchy po elipsie mogą być stabilne, ale wymagają, podobnie jak ruchy po okręgu, by jedno z ciał było znacznie cięższe od pozostałych. Ponadto ekscentryczność orbity (czyli spłaszczenie elips) nie może być zbyt duża.



*Pisząc limeryk o Teresie  
Pogrążysz się w głębokim stresie. . .*

Wisława Szymborska

Pisząc limeryk o torusie,  
Chciałem spleść w torus ze trzy strusie,  
Ale tymczasem moja żona  
Mówi, że bierze się pytona  
I ogon w paszczę wtyka mu się.

Był szef izby wytrzeźwień w Piwnicznej  
Specjalistą od algebraicznej  
Sztuki doprowadzania  
Dowolnego równania  
Do postaci jak wół kanonicznej.

W zeszłym roku na jednej z plaż w Gdyni  
Na najkrótsze raz konkurs był mini.  
Szans nie miało tam wcale  
Mini na cztery cale,  
Bowiem minus pięć ma zwyciężczyni.

Operator w dowolnej przestrzeni  
Miał ambicję, że cośkolwiek zmieni,  
Lecz mu wbili do głowy,  
Że jest tożsamościowy.  
Zwątpił w siebie i teraz się leni.

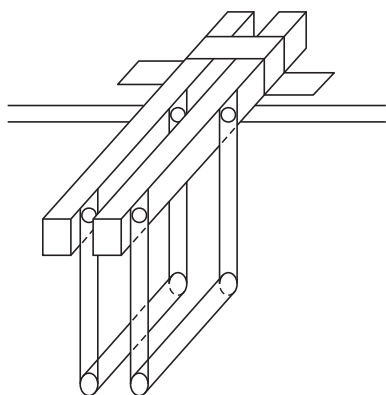
Informatyk spod Leśnej Podkowy  
Zwykł stosować zbiór liczb szesnastkowy.  
Nawet myśląc o seksie  
Wszystko liczył on w heksie.  
A poza tym zupełnie był zdrowy.

Geometra, mieszkaniec Gołdapi  
Na dziewczyny bez przerwy się gapi,  
Wypatrując dowodu,  
Że to, co mają z przodu  
Ma w obwodzie  $2r \times 2\pi$ .

Przemysław KICIAK







Rys. 5. Zawieszenie dwóch rurek obok siebie.

przepływu prądu, to odchylony kciuk wskaże nam kierunek działania siły elektrodynamicznej (rys. 4). Należy przy tym pamiętać, że linie pola magnetycznego wychodzą z bieguna północnego i wchodzą do południowego, a umowny kierunek przepływu prądu jest od bieguna dodatniego do ujemnego. Korzystając z tej reguły, możemy wyznaczyć bieguny posiadanego magnesu.

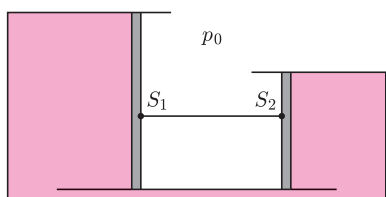
Jeżeli do stołu przykleimy listewkę z drugą rurką tak, żeby rurki były równoległe, a odległość między nimi wynosiła 1–2 cm, to będziemy mogli zbadać wzajemne oddziaływanie przewodników, w których płynie prąd elektryczny (rys. 5). Powinniśmy przy tym zauważyć, że rurki przyciągają się, kiedy kierunki przepływu prądu w nich są zgodne, a odpychają się, gdy są przeciwnie.

Dysponując większą liczbą baterii płaskich (4–5 sztuk) połączonych szeregowo plus do minusa i zasilających pojedynczą rurkę, możemy zauważyć odchylenie rurki w polu magnetycznym Ziemi. Należy przy tym zwrócić uwagę, żeby w pobliżu rurki nie znalazły się przypadkowo jakieś magnesy lub przedmioty wykonane ze stali. Korzystając z reguły lewej dłoni, będziemy mogli wówczas wyznaczyć kierunek linii składowej ziemskiego pola magnetycznego prostopadłej do rurki.



## Zadania

Redaguje Ewa CZUCHRY



**F 703.** Dwa cylindry o przekrojach  $S_1$  i  $S_2$  zostały połączone małą rurką (rysunek). Cylindry są zamknięte sztywno połączonymi tłokami. Objętość gazu znajdującego się między tłokami była początkowo równa  $V$ , a ciśnienie wewnątrz układu równe atmosferycznemu  $p_0$ . Na jaką odległość trzeba przesunąć tłok, tak aby ciśnienie wewnątrz cylindrów zmieniło się do wartości  $p$ ? Temperaturę gazu uznajemy za stałą, a tarcie zaniedbujemy.

Rozwiązanie na str. 4

**F 704.** W stojącym pionowo naczyniu cylindrycznym wypełnionym powietrzem, znajdują się dwa jednakowe masywne tłoki. Odległość między nimi oraz wysokość dolnego tłoka od poziomu dna naczynia wynoszą  $d = 5$  cm. Ciśnienie między tłokami jest dwa razy większe od atmosferycznego  $p_0$ . Dociążamy górny tłok tak, aby znalazł się on na wysokości poprzednio zajmowanej przez ten dolny. Na jakiej wysokości znajdzie się dolny tłok? Temperaturę powietrza uznajemy za stałą, a tarcie zaniedbujemy.

Rozwiązanie na str. 8

Redaguje Waldemar POMPE

**M 1186.** Każdy wyraz ciągu  $a_1, a_2, \dots$  jest kwadratem liczby całkowitej dodatniej. Ponadto dla każdej liczby naturalnej  $n$ , różnica  $a_{n+1} - a_n$  jest liczbą pierwszą lub kwadratem liczby pierwszej. Wykazać, że ciąg  $a_1, a_2, \dots$  jest skończony.

Rozwiązanie na str. 5

**M 1187.** Czy istnieje taki ostrosłup o podstawie pięciokąta wypukłego, którego każda ściana boczna jest trójkątem prostokątnym?

Rozwiązanie na str. 24

**M 1188.** Każdy wyraz rosnącego ciągu  $x_1 < x_2 < \dots$  jest liczbą całkowitą dodatnią. Ponadto ciąg ten spełnia nierówności

$$x_{n+1} \leq 2n \quad \text{dla } n = 1, 2, \dots$$

Wykazać, że każdą liczbę całkowitą dodatnią można przedstawić w postaci różnicy pewnych dwóch wyrazów tego ciągu.

Rozwiązanie na str. 18



## Skojarzenia



Start informatyki w *Delcie* dobrze zbiegł się w czasie z innym wydarzeniem, a mianowicie z dziewiętnastą Międzynarodową Olimpiadą Informatyczną, która w dniach 15–22 sierpnia 2007 odbyła się w Zagrzebiu. Zawody Olimpiady polegają na rozwiązywaniu zadań programistycznych w czasie dwóch pięciogodzinnych sesji. W tym roku w Olimpiadzie uczestniczyło ponad 280 uczniów z kilkudziesięciu krajów świata, a zwyciężył Polak – **Tomasz Kulczyński**, absolwent VI LO im. J. i J. Śniadeckich w Bydgoszczy. Tomasz zdobył 574 punkty na 600 możliwych, wyprzedzając drugiego zawodnika o 19 punktów. Świetny wynik – 504 punkty i szóste miejsce – uzyskał inny reprezentant Polski, **Marcin Andrychowicz**. Gratulujemy!

Wynik Tomka jest fantastyczną powtórką wyniku z poprzedniego roku. Wtedy to, podczas osiemnastej Olimpiady, odbywającej się w Meksyku, po raz pierwszy w historii pierwsze miejsce zajął nasz inny reprezentant – **Filip Wolski**, który zdobył 480 punktów. Pozostaje wierzyć w przyszłowie, że „do (co najmniej) trzech razy sztuka” i czekać z niecierpliwością na wyniki 20. Olimpiady, która zagości w przyszłym roku w Egipcie.

Skoro już mowa o Filipie, nie sposób nie wspomnieć o jego innym zwycięstwie. W marcu tego roku drużyna Uniwersytetu Warszawskiego w składzie: **Filip Wolski**, **Marcin Pilipczuk** i **Marek Cygan**, zwyciężyła w finale Akademickich Mistrzostw Świata w Programowaniu Zespołowym ACM, które odbyły się w Tokio. Jest to najstarszy i najbardziej prestiżowy konkurs programistyczny na świecie, a jego specyfika jest zupełnie inna niż Olimpiady — w zawodach startują trzyosobowe zespoły, z których każdy ma do dyspozycji tylko jeden komputer. Polski zespół pokonał w finale 87 reprezentacji najlepszych uczelni z całego świata, rozwiązując 8 z 10 zadań – więcej niż którakolwiek z pozostałych drużyn.

Nie był to pierwszy sukces Polaków w tych zawodach. W finałach roku 2006 drugie miejsce zajęła drużyna Uniwersytetu Jagiellońskiego w składzie: **Arkadiusz Pawlik**, **Bartosz Walczak** i **Paweł Walter**. Studenci z Krakowa ulegli minimalnie jedynie drużynie z Rosji – rozwiązyli tyle samo zadań, ale w nieco gorszym czasie. W roku 2005 piąte miejsce zajęła ekipa Uniwersytetu Wrocławskiego. Natomiast w roku 2003 Polakom po raz pierwszy udało się wspiąć na najwyższy stopień podium. Wyczynu tego dokonała drużyna Uniwersytetu Warszawskiego w składzie: **Tomasz Czajka**, **Andrzej Gąsienica-Samek** i **Krzysztof Onak**. To historyczne zwycięstwo odbiło się wówczas bardzo szerokim echem i na pewno zwróciło światową uwagę na wysoki poziom polskiej informatyki.

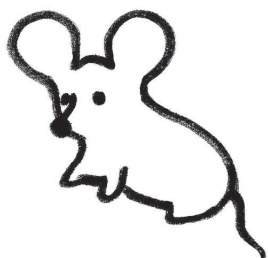
Wspomniany Tomasz Czajka jest zresztą znany za sprawą konkursu TopCoder. W TopCoderze średnio raz na tydzień organizowane są tzw. mecze, w których zawodnicy startują przez Internet, a kilka razy do roku odbywają się turnieje „na żywo”. Jest to dynamiczny i emocjonujący konkurs – każdy zawodnik ma niewiele ponad godzinę na rozwiązanie trzech zadań, a potem może jeszcze próbować „wyłożyć” programy przeciwników. W finale konkursu TopCoder Open Tomek kilkakrotnie zajmował pierwsze miejsca, a poza tym długo utrzymywał się na czele internetowego rankingu w kategorii *Algorithms*.

Na stronie TopCodera można poza tym znaleźć wiele innych rankingów. W chwili obecnej w kategorii *Kraje* Polska zajmuje drugie miejsce. Natomiast w kategorii *Szkoły* (w której wynik uczelni jest wyliczany na podstawie wyników reprezentujących ją indywidualnie zawodników) pierwsze miejsce zajmuje Uniwersytet Warszawski. Od października studentem tegoż Uniwersytetu jest nie kto inny, jak... Tomasz Kulczyński. I tak oto koło „pierwszych miejsc” się zamyka.

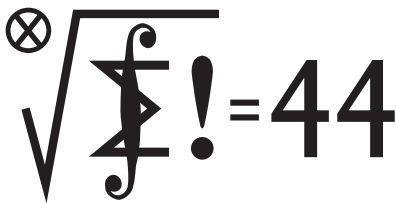
Michał ADAMASZEK

PS. I jeszcze informacja „z ostatniej chwili” – do swoich sukcesów Filip Wolski dorzucił jeszcze jeden: objął redakcję prestiżowego Informatycznego Kącika Olimpijskiego w naszym czasopiśmie (patrz *Delta* 10/2007, strona 19 i strona 10 w tym numerze).

<http://olympiads.win.tue.nl/ioi/>  
<http://icpc.baylor.edu/icpc/>  
<http://www.topcoder.com/tc>



# Klub 44



Termin nadsyłania rozwiązań: 31 I 2008

Czołówka ligi zadaniowej

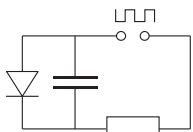
**Klub 44 M**

po uwzględnieniu ocen rozwiązań zadań  
537 ( $WT = 2,42$ ) i 538 ( $WT = 1,50$ )  
z numeru 3/2007

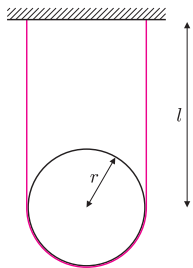
Tomasz Wietecha	– Tarnów	47,09
Andrzej Daniluk	– Warszawa	42,70
Dariusz Kurpiel	– Posada	
	Zarszyn	42,51
Witold Bednarek	– Łódź	38,60
Krzysztof Kamiński	– Pabianice	38,05
Grzegorz Karpowicz	– Wrocław	37,78

I znów moment godny odnotowania:

Tomasz Wietecha zgromadził 44 punkty już po raz siódmy!



Rys. 1



Rys. 2



## Liga zadaniowa Wydziału Matematyki, Informatyki i Mechaniki, Wydziału Fizyki Uniwersytetu Warszawskiego i Redakcji *Delty*

Skrót regulaminu

Każdy może nadsyłać rozwiązania zadań z numeru  $n$  w terminie do końca miesiąca  $n + 2$ . Szkice rozwiązań zamieszczamy w numerze  $n + 4$ . Można nadsyłać rozwiązania czterech, trzech, dwóch lub jednego zadania (każde na oddzielnej kartce), można to robić co miesiąc lub z dowolnymi przerwami. Rozwiązania zadań z matematyki i z fizyki należy przysyłać w oddzielnych kopertach, umieszczając na kopercie dopisek: **Klub 44 M** lub **Klub 44 F**. Oceniamy zadania w skali od 0 do 1 z dokładnością do 0,1. Ocenę mnożymy przez współczynnik trudności danego zadania:  $WT = 4 - 3S/N$ , gdzie  $S$  oznacza sumę ocen za rozwiązania tego zadania, a  $N$  – liczbę osób, które nadesłały rozwiązanie choćby jednego zadania z danego numeru w danej konkurencji (**M** lub **F**) – i tyle punktów otrzymuje nadsyłający. Po zgromadzeniu 44 punktów, w dowolnym czasie i w którejkolwiek z dwóch konkurencji (**M** lub **F**), zostaje on członkiem **Klubu 44**, a nadwyżka punktów jest zaliczana do ponownego udziału. Trzykrotne członkostwo – to tytuł **Weterana**. Szczegółowy regulamin został wydrukowany w numerze 2/2002 oraz znajduje się na stronie <http://www.mimuw.edu.pl/delta/regulamin.html>.

### Zadania z matematyki nr 549, 550

Redaguje Marcin E. KUCZMA

**549.** Dany jest trójkąt  $ABC$  o bokach długości  $|AB| > |BC| > |CA|$ . Odcinek  $CK$  jest jego wysokością; punkty  $M$  i  $N$  są odpowiednio środkami boków  $AB$  i  $AC$ . Okrąg wpisany w trójkąt  $ABC$  jest styczny do boków  $AB$  i  $AC$  odpowiednio w punktach  $P$  i  $Q$ . Udowodnić, że środek okręgu wpisanego w trójkąt  $KMN$  leży na prostej  $PQ$ .

**550.** Dla każdej liczby rzeczywistej  $a$  wyznaczyć kres dolny zbioru tych liczb rzeczywistych  $x$ , które spełniają nierówność  $[x] \cdot \{x\} \geq a$ . (Symbol  $\{x\}$  oznacza tu liczbę  $x - [x]$ ).

Zadanie 550 zaproponował pan Paweł Najman z Jaworzna.

### Zadania z fizyki nr 446, 447

Redaguje Jerzy B. BROJAN

**446.** Do źródła napięcia o przebiegu prostokątnym (tzn.  $U = +U_0$  w ciągu czasu  $T$ , następnie  $U = -U_0$  w takim samym przedziale czasu) przyłączono obwód składający się z opornika o oporze  $R$ , kondensatora o pojemności  $C$  oraz diody doskonale przewodzącej w kierunku przewodzenia i całkowicie nieprzewodzącej w kierunku zaporowym (schemat obwodu – rys. 1). Jak długo płynie prąd przez diodę w ciągu jednego okresu zmian napięcia?

**447.** Jednorodny walec o promieniu  $r$  wisi na dwóch pętlach lekkiej i nierozciągliwej nitki (zob. rys. 2), a odległość osi walca od poziomu zawieszenia jest równa  $l$ . Odległość końców każdej z nitek jest równa  $2r$ . Obliczyć częstotliwość małych drgań takiego wahadła – oś walca pozostaje równoległa do jej kierunku w położeniu równowagi.

## Można też inaczej

W artykule *Równanie Pitagorasa w kongruencjach* w poprzednim numerze *Delty* autorzy dowodzą, że równanie Pitagorasa

$$x^2 + y^2 = z^2$$

ma dokładnie  $p^2$  rozwiązań w ciele  $\mathbb{Z}_p$ . Oto elementarny dowód: najpierw zauważmy, że równanie Pitagorasa, które można równoważnie zapisać w postaci:

$$(1) \quad x^2 = (z - y)(z + y)$$

ma nad ciałem charakterystyki różnej od 2 tyle samo rozwiązań, co równanie:

$$(2) \quad a^2 = bc.$$

Istotnie, jeśli  $(x, y, z)$  jest rozwiązaniem równania (1), to  $(a, b, c) := (x, z - y, z + y)$  jest rozwiązaniem równania (2) i odwrotnie: rozwiązanie  $(a, b, c)$

równania (2) daje rozwiązanie  $(x, y, z) := (a, \frac{c-b}{2}, \frac{c+b}{2})$  równania (1). Te przekształcenia pomiędzy trójkami  $(x, y, z)$  rozwiązań pierwszego równania i trójkami  $(a, b, c)$  rozwiązań drugiego równania są wzajemnie odwrotne i zadają bijekcję zbiorów rozwiązań.

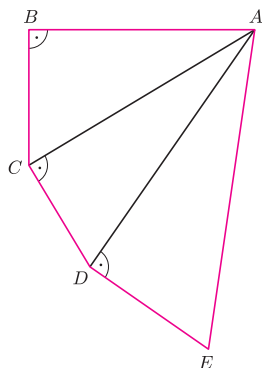
Rozwiązania równania (2) w  $\mathbb{Z}_p$  łatwo zliczyć. Każda para  $(a, b)$ , gdzie  $b \neq 0$  wyznacza jedyną wartość  $c = a^2 b^{-1}$ , przy której równanie jest spełnione. Takich par  $(a, b)$  jest  $p(p-1)$ . Ponadto mamy rozwiązania postaci  $(0, 0, c)$  dla dowolnego  $c$ . Łącznie rozwiązań jest więc  $p(p-1) + p = p^2$ . To samo rozumowanie pokazuje, że w ciele  $\mathbb{F}_p^m$  (dla  $p \neq 2$ ) rozwiązań równania Pitagorasa jest  $p^m(p^m - 1) + p^m = p^{2m}$ .

Michał ADAMASZEK



### Rozwiązanie zadania M 1187.

Taki ostrosłup istnieje. Niech  $ABCDE$  będzie pięciokątem wypukłym złożonym z trzech trójkątów prostokątnych, jak pokazano na rysunku.



Niech ponadto  $S$  będzie takim punktem w przestrzeni, którego rzut prostokątny na płaszczyznę pięciokąta  $ABCDE$  pokrywa się z punktem  $A$ . Wówczas  $\sphericalangle SAB = \sphericalangle SAE = 90^\circ$ , jak również  $\sphericalangle SBC = \sphericalangle SCD = \sphericalangle SDE = 90^\circ$ . Zatem każda ściana boczna ostrosłupa pięciokątnego  $ABCDE$  jest trójkątem prostokątnym.

## Patrz w niebo

Najpotężniejsze ziemskie akceleratory są w stanie nadawać cząstkom elementarnym energię rzędu TeV, czyli  $10^{12}$  eV. W promieniowaniu kosmicznym obserwuje się (fakt, że bardzo rzadko) cząstki o energii setek milionów TeV, sięgającej  $3 \cdot 10^{20}$  eV. Ponieważ  $1 \text{ eV} = 1,6 \cdot 10^{-19} \text{ J}$ , to łatwo sprawdzić, że jest to energia 50 J, czyli pojedynczy proton ma energię np. kilogramowego ciężarka lecącego z prędkością 10 m/s! Z samego faktu występowania takich cząstek wynika, że ich źródło musi być – w skali kosmicznej – niezbyt odległe. Gdyby bowiem było bardzo odległe, to cząstki zdążyłyby stracić część energii w wyniku oddziaływania z kwantami promieniowania relikтового, wypełniającymi cały Wszechświat. Druga istotna uwaga: mając taką energię, proton właściwie nie zauważa już pola magnetycznego naszej Galaktyki, które wyraźnie odchyłają tor cząstek o niższych energiach. O ile więc nie da się określić, skąd pochodzą cząstki „zwykłego” promieniowania kosmicznego, to w przypadku tych najbardziej energetycznych można o to się pokusić.

Grupa astronomów z Princeton University kilka lat temu przebadła nieliczne przypadki zaobserwowania cząstek promieniowania kosmicznego o makroskopowych energiach i jako ich przypuszczalne źródła wytypowała 12 galaktyk zawierających w centrum supermasywną czarną dziurę i będących najprawdopodobniej pozostałościami po kwazarach. Na podstawie obserwacji 40 superenergetycznych cząstek, zarejestrowanych przez detektor promieniowania kosmicznego zainstalowany pod Tokio, badacze doszli do wniosku, że ich źródłami mogą być cztery galaktyki położone w Wielkiej Niedźwiedzicy i w okolicy: NGC 3610, 3613, 4589 i 5322. Wysłunęto hipotezę, że rotacja tych czarnych dziur może pośrednio być przyczyną rozpędzania cząstek. Wydaje się mianowicie, że pole magnetyczne „wleczone” przez pole grawitacyjne wirującej czarnej dziury może indukować potężne pole elektryczne, zdolne z kolei niesłychanie rozpędzać cząstki naładowane. Obecnie statystyka tych rzadkich obserwacji jest jednak zbyt uboga, by tę hipotezę wyraźnie potwierdzić lub obalić.

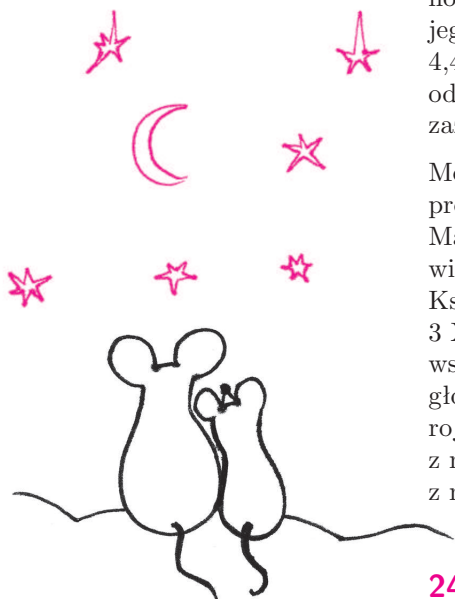
Tomasz KWAST

## Listopad

W listopadowe wieczory ledwo-ledwo widać nad południowym horyzontem północną część Rzeźbiarza, gwiazdozbioru nieba południowego. Warto jednak o nim wspomnieć dlatego, że w nim leży południowy biegun Galaktyki (północny leży w Warkoczu Bereniki, w gwiazdozbiornie widocznym wysoko na naszym niebie w majowe wieczory; teraz jest on wieczorem pod północnym horyzontem). Droga Mleczna przebiega w listopadzie wieczorem ze wschodu na zachód dość blisko zenitu, podczas gdy w maju nisko nad północnym horyzontem. Sam Rzeźbiarz jest gwiazdozbiorem małym i mało okazałym; jego najjaśniejsza gwiazda, alfa, to niebiesko-biały olbrzym o jasności zaledwie 4,4 mag, leżący w odległości 80 pc od nas. Liczne tam galaktyki (o jasnościach od 7 mag) byłyby widoczne już przez niewielką lunetę, gdyby nie kiepska zazwyczaj przezroczystość atmosfery w pobliżu horyzontu.

Merkury znajdzie się 8 XI najdalej, tj. o  $19^\circ$ , na zachód od Słońca i można próbować go dostrzec o świcie. Wenus jest w Pannie, również więc widać ją rano. Mars jest w Bliźniętach i widać go przez całą noc. Jowisz jest zbyt blisko Słońca, więc go nie widać, a Saturn we Lwie i widać go w drugiej połowie nocy. Nów Księżyca wypada 9 XI, a pełnia 24 XI. Księżyc zakryje dwa razy Regulusa: 3 XI zobaczą to zjawisko mieszkańcy środkowej Ameryki i 30 XI mieszkańcy wschodniej Azji i Australii, oraz zakryje Antaresa 11 XI, co będzie widoczne głównie z południowego Pacyfiku. W listopadzie można spodziewać się trzech rojów meteorów. Taurydy mają maksimum 7 XI, ale to rój raczej słaby, Leonidy z maksimum 15 XI mogą dać około 10 błysków na godzinę; Andromedydy z maksimum 23 XI to rój bardzo słaby.

T. K.



# Losowe rozbicia odcinka, proces Poissona i paradoks czasu oczekiwania

Rafał SZTENCEL

Rozkład wykładniczy ma pewną godną uwagi własność. Wyobraźmy sobie, że opisuje on czas oczekiwania  $T$  na pewne zjawisko, zatem

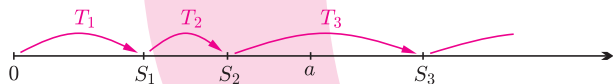
$$P(T > t) = e^{-\lambda t}, \quad t \geq 0, \lambda > 0.$$

Podaliśmy ogon rozkładu, czyli szansę, że czas oczekiwania przekroczy  $t$ . Przypuśćmy, że czekamy już  $s$  jednostek czasu. Jaka jest szansa, że poczekamy jeszcze co najmniej  $t$ ? Obliczamy za pomocą prawdopodobieństwa warunkowego:

$$\begin{aligned} P(T > s + t | T > s) &= \frac{P(T > s + t, T > s)}{P(T > s)} = \\ &= \frac{P(T > s + t)}{P(T > s)} = \frac{e^{-\lambda(s+t)}}{e^{-\lambda s}} = \\ &= e^{-\lambda t} = P(T > t), \quad s, t \geq 0. \end{aligned}$$

Rezultat nie zależy od  $s$ , dlatego powyższą własność określa się jako brak pamięci dla rozkładu wykładniczego. Wynika z niej, że nawet gdy czekamy już bardzo długo, średni dalszy czas oczekiwania jest i tak równy  $ET = 1/\lambda$ .

Rozpatrzmy teraz ciąg niezależnych czasów oczekiwania  $T_1, T_2, \dots$ , o tym samym rozkładzie co  $T$ . Mogą to być, na przykład, czasy oczekiwania na zgłoszenia do centrali telefonicznej. Proces startuje w chwili 0, a kolejne zgłoszenia nadchodzą w momentach  $S_n = T_1 + \dots + T_n$ . W chwili  $a$  rozpoczynamy obserwację procesu. Jaki jest średni czas oczekiwania  $m(a)$  na najbliższe zgłoszenie?



Możliwe są dwie odpowiedzi: (1) z własności braku pamięci wynika, że  $m(a) = 1/\lambda$ , bo nie jest istotne, kiedy miało miejsce ostatnie zgłoszenie poprzedzające  $a$ ; (2) ale przecież średni odstęp między zgłoszeniami jest równy  $1/\lambda$ , i skoro punkt  $a$  znajduje się wewnątrz odcinka o średniej długości  $1/\lambda$ , to można się spodziewać, że  $m(a)$  będzie z grubsza dwukrotnie mniejsze.

Zanim rozstrzygniemy ten paradoks, mała dygresja o losowym rozbiciu okręgu. Jeśli wybierzemy losowo (dokładniej: zgodnie z rozkładem jednostajnym) i niezależnie punkty  $A$  i  $B$  na okręgu o długości 1, to średnie długości łuków  $AB$  i  $BA$  będą równe  $1/2$ . A jeśli przed losowaniem ustalimy punkt  $P$ ? Teraz łuk zawierający ten punkt będzie miał średnią długość  $2/3$ . Istotnie, trzy punkty wybrane losowo i niezależnie z okręgu dzielą go na łuki o średniej długości  $1/3$ . Wystarczy teraz obrócić okrąg tak, by np. pierwszy wylosowany punkt pokrył się z  $P$ .

Odpowiednio wykształcona intuicja probabilistyczna powinna podpowiedzieć, że dłuższy przedział ma większe szanse przykrycia punktu  $P$  niż krótszy. Można pokazać, że rozkład długości łuku  $AB$  jest jednostajny, ale rozkład długości łuku zawierającego punkt  $P$  ma gęstość  $2x$  dla  $x \in [0, 1]$ .



Analogiczne zjawisko powoduje, że argumenty (1) i (2) nie są sprzeczne. Przedział zawierający punkt  $a$  jest średnio dłuższy niż  $1/\lambda$  i odpowiedź (1) nie prowadzi do paradoksu, bowiem rzeczywiście średni czas oczekiwania na najbliższe zgłoszenie po chwili  $a$  jest równy  $1/\lambda$ .

Natomiast średni czas od ostatniego zgłoszenia poprzedzającego  $a$  do  $a$  jest równy  $\frac{1}{\lambda}(1 - e^{-\lambda a})$ , więc dla dużych  $a$  przedział nakrywający punkt  $a$  ma długość niewiele mniejszą niż  $2/\lambda$ . Oto szkic dowodu.

Po pierwsze, można wyliczyć gęstość  $g_n$  i dystrybuantę  $F_n$  zmiennej losowej  $S_n$ :

$$g_n(x) = \frac{\lambda^n x^{n-1}}{(n-1)!},$$

$$F_n(x) = 1 - e^{-\lambda x} \left( 1 + \lambda x + \dots + \frac{(\lambda x)^{n-1}}{(n-1)!} \right), \quad x \geq 0.$$

Stąd liczba  $N_a$  zgłoszeń w przedziale  $[0, a]$  ma rozkład Poissona z parametrem  $\lambda a$ :

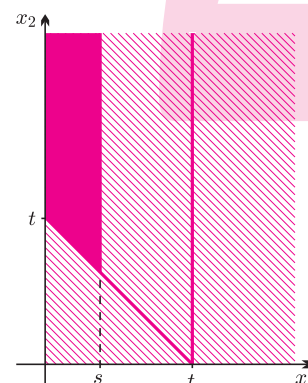
$$\begin{aligned} P(N_a = k) &= P(S_k \leq a, S_{k+1} > a) = \\ &= P(\{S_k \leq a\} \setminus \{S_{k+1} \leq a\}) = \\ &= F_k(a) - F_{k+1}(a) = \frac{(\lambda a)^k}{k!} e^{-\lambda a}. \end{aligned}$$

Po drugie, jeśli wiemy, że w przedziale  $[0, a]$  było  $k$  zgłoszeń, to układają się one tak, jak  $k$  punktów rzuconych losowo i niezależnie<sup>1</sup>. Dlatego przedział  $[0, a]$  zostanie podzielony na  $(k+1)$  przedziałów o średniej długości  $a/(k+1)$ . Łatwo to zobaczyć, gdy odwołamy się do losowego rozbicia okręgu punktami  $X_1, X_2, \dots, X_{k+1}$ . Jasne jest, że wszystkie łuki mają równe średnie długości. Po rozcięciu okręgu w punkcie  $X_{k+1}$  otrzymujemy losowe rozbicie odcinka.

Teraz za pomocą uogólnienia wzoru na prawdopodobieństwo całkowite obliczamy średni odstęp  $a$  od poprzedzającego zgłoszenia. Jest on równy

$$\begin{aligned} \sum_{k=0}^{\infty} \frac{a}{k+1} \frac{(\lambda a)^k}{k!} e^{-\lambda a} &= \frac{1}{\lambda} \sum_{k=0}^{\infty} \frac{(\lambda a)^{k+1}}{(k+1)!} e^{-\lambda a} = \\ &= \frac{1}{\lambda} (e^{\lambda a} - 1) e^{-\lambda a} = \frac{1}{\lambda} (1 - e^{-\lambda a}). \end{aligned}$$

Podaliśmy tu jedną z możliwych konstrukcji procesu Poissona. Z formalnego punktu widzenia jest on rodziną zmiennych losowych  $(N_t)_{t \geq 0}$ , liczących ilość zgłoszeń do chwili  $t$ .



<sup>1</sup>Dla  $k = 1$  można wykazać, że  $P(X_1 < s | X_1 \leq t, X_1 + X_2 > t) = \frac{s}{t}$ , gdy  $0 \leq s \leq t$  na podstawie rysunku (patrz obok); wystarczy skorzystać z faktu, że gęstość łącznego rozkładu  $(X_1, X_2)$ , wyrażająca się w pierwszej ćwiartce wzorem  $\lambda^2 e^{-\lambda(x_1+x_2)}$ , jest stała na odcinkach  $x_1 + x_2 = c$ ,  $x_1, x_2 \geq 0$ . W wyższych wymiarach idea dowodu jest ta sama.