Gra w zgadywanie

Jakub RADOSZEWSKI

Jaś i Małgosia grają w następującą *Grę w zgadywanie*. Na początku gry Jaś wybiera sobie liczbę $x \in \{1, \ldots, n\}$. Małgosia będzie próbować odgadnąć x, zadając Jasiowi pytania postaci: „Czy $x \leqslant a$?”, na które Jaś odpowiada „Tak” lub „Nie”. Małgosia dąży przy tym do zminimalizowania liczby zadanych pytań, natomiast Jasiowi zależy na tym, żeby Małgosia musiała ich zadać jak najwięcej.

No dobrze, ale co to tak naprawdę znaczy, że Małgosia chce zminimalizować liczbę zadanych pytań? Powiedzmy, że dla pewnego k mamy następujące dwie strategie:

- Jeżeli $x \neq \lfloor n/2 \rfloor$, to wartość x zostaje odgadnięta za pomocą co najwyżej k pytań, lecz odgadnięcie wartości x w przypadku, gdy $x = \lfloor n/2 \rfloor$, wymaga zadania $k+2$ pytań.
- Odgadnięcie wartości x wymaga zawsze zadania $k+1$ pytań.

Którą z nich Małgosia uzna za lepszą?

Uzasadnienie takiego a nie innego wyboru strategii znajduje się w dalszej części tekstu.

Z naszego punktu widzenia lepsza będzie druga z powyższych strategii. Ogólniej, będziemy poszukiwać takiej strategii gry dla Małgosi, przy której liczba pytań koniecznych do odgadnięcia wartości x *w najgorszym przypadku* będzie najmniejsza możliwa. W przypadku gry w zgadywanie przez najgorszy przypadek rozumiemy najgorszą z punktu widzenia danej strategii wartość x. Jest to zarazem jedyna rzecz, na którą Jaś ma wpływ w opisanej grze, więc jego celem będzie jak najbardziej złośliwy wobec Małgosi wybór x.

Intuicja podpowiada, że całkiem niezłą strategię gry dla Małgosi można skonstruować na podstawie *wyszukiwania binarnego*. Dokładniej, w pierwszym pytaniu wybieramy $a = \lfloor n/2 \rfloor$. W zależności od odpowiedzi na to pytanie będziemy się odtąd zajmować zbiorem $\{1, \ldots, a\}$ lub $\{a+1, \ldots, n\}$. W pierwszym przypadku kolejną wartością, o którą spytamy, będzie $\lfloor a/2 \rfloor$, w drugim zaś środkowy całkowity element przedziału $[a+1, n]$ itd. Można łatwo sprawdzić, że liczba pytań, jakie będziemy musieli zadać w najgorszym przypadku, jest przy tej strategii równa $\lceil \log n \rceil$ (gdyż po każdym pytaniu długość przedziału poszukiwań zostaje mniej więcej przepołowiona).

Czy jest to z naszego punktu widzenia strategia optymalna? Okazuje się, że tak! Otóż w dowolnej strategii, po otrzymaniu odpowiedzi na pytanie „Czy $x \leqslant a$?” wiemy, że x znajduje się w zbiorze $\{1, \ldots, a\}$ lub też w $\{a+1, \ldots, n\}$. Od tego momentu możemy zadawać kolejne pytania, ograniczając się do jednego z tych dwóch przedziałów; nową wartością n przed drugim pytaniem będzie więc a lub $n-a$. Ponieważ nas interesuje najgorszy możliwy przypadek, to możemy założyć, że x znajduje się w dłuższym spośród tych dwóch przedziałów. Łatwo zauważyć, że właśnie w wyszukiwaniu binarnym wartość $\max(a, n-a)$ jest najmniejsza możliwa.

Zauważmy, że w drugim z przypadków liczby ze zbioru $\{a+1, \ldots, n\}$ możemy utożsamiać z liczbami $\{1, \ldots, n-a\}$.

Mamy już strategię dla Małgosi, ale jak podpowiedzieć coś Jasiowi? Ponieważ Jaś nie powinien zakładać żadnej konkretnej strategii Małgosi, to może mu być strasznie ciężko odgadnąć, jakie x sprawi jej najwięcej kłopotów. Z tego względu najlepszym rozwiązaniem dla Jasia będzie… drobne oszustwo. Otóż Jaś wcale nie potrzebuje wymyślać sobie na początku wartości x; wystarczy, że na pytania Małgosi będzie zawsze odpowiadał tak, żeby długość pozostającego jej przedziału poszukiwań była możliwie największa. Jeżeli Jaś będzie przy tym odpowiednio ostrożny (czyli nigdy nie udzieli odpowiedzi sprzecznej z poprzednimi), to jego oszustwo nigdy nie zostanie wykryte! Zauważmy też, że taka strategia Jasia wymusza na Małgosi branie pod uwagę zawsze najgorszego możliwego przebiegu gry, co uzasadnia nasz wcześniejszy wybór kryterium porównywania strategii.

Jasiowi i Małgosi znudziła się już prościutka gra w zgadywanie, więc wymyślili oni jej nową wersję, która zwłaszcza Jasiowi bardzo się spodobała: teraz Jaś

Rozwiązanie zadania F 731.
Z prawa Faradaya mamy

$$|\mathcal{E}_{\text{ind}}| = B\frac{\Delta S}{\Delta t} = Bl^2\omega^2/2.$$

Z porównania mocy prądu i układu otrzymujemy

$$Fv = Fl\omega = \frac{\mathcal{E}_{\text{ind}}^2}{R},$$

stąd

$$F = \frac{B^2 l^3 \omega}{4R}.$$

może kłamać przy odpowiadaniu na pytania! Tym razem Małgosia nie była jednak w stanie skutecznie odgadnąć żadnej wartości x, więc dzieci postanowiły ograniczyć swobodę Jasia – może on skłamać co najwyżej raz w ciągu jednej gry. Małgosia wymyśliła już sposób modyfikacji strategii wyszukiwania binarnego, dzięki któremu działa ona także dla nowej gry. Otóż pytania można zadawać tak samo jak poprzednio, ale żeby nie dać się wpuścić w maliny, wystarczy każde z nich dwukrotnie powtórzyć. Jeżeli odpowiedzi będą takie same, to Jaś nie skłamał, a jeżeli różne, to Małgosia zadaje jeszcze trzecie pytanie, żeby poznać prawdziwą odpowiedź i, co więcej, odtąd już wie, że Jaś nie może więcej kłamać, więc może ona zastosować dokładnie strategię wyszukiwania binarnego. Można obliczyć, że wartość x zostanie w ten sposób odgadnięta zawsze w co najwyżej $2 \cdot \lceil \log n \rceil + 1$ ruchach. Dzieci zaczęły się jednak zastanawiać, czy jest to strategia optymalna dla tej gry.

W opisanej strategii dla $n = 5$ może być konieczne zadanie 7 pytań. Czy potrafisz wskazać strategię, w której w tym przypadku 6 pytań zawsze wystarczy? Odpowiedź na to pytanie może wymagać nieco pokombinowania.

Okazuje się, że nie, a pierwszą wartością n, dla której opisana strategia może wymagać więcej pytań niż optymalna, jest 5. Nie zrazimy się jednak początkowym niepowodzeniem i skonstruujemy optymalną strategię również i dla tej gry, choć tym razem może nam się przydać do tego pomoc komputera.

Załóżmy więc, że Małgosia zadała pytanie „Czy $x \leqslant a$?” i uzyskała na nie odpowiedź „Tak”. Co Małgosia może z tej odpowiedzi wywnioskować o wartości x? W sumie… niestety nic. Jeżeli Jaś powiedział prawdę, to $x \leqslant a$, ale równie dobrze może zachodzić $x > a$, jeżeli tylko Jaś skłamał. Musi jednak istnieć możliwość wywnioskowania czegokolwiek, gdyż w przeciwnym razie gra by się nigdy nie skończyła. Okazuje się, że na podstawie odpowiedzi Jasia można wysnuć wnioski nie tyle o wartości x, ale o liczbie dotychczasowych kłamstw Jasia w zależności od tej wartości. Faktycznie, jeżeli $x \leqslant a$, to Jaś jeszcze ani razu nie skłamał, a w przeciwnym przypadku skłamał dokładnie raz.

Następnie Małgosia zadaje Jasiowi drugie pytanie „Czy $x \leqslant b$?”, tym razem powiedzmy dla $b > a$. Jeżeli Jaś na to pytanie również odpowiedział „Tak”, to wiadomo, że:

- jeżeli $x \leqslant a$, to Jaś jeszcze ani razu nie skłamał;
- jeżeli $a < x \leqslant b$, to Jaś skłamał dokładnie raz;
- jeżeli $x > b$, to Jaś skłamał dokładnie dwa razy; to jest jednakże niemożliwe, gdyż Jaś ma do dyspozycji tylko jedną możliwość skłamania.

Jeżeli zaś Jaś na drugie pytanie Małgosi odpowiedział przecząco, to można z tego wywnioskować, że $x \leqslant a$ lub $x > b$ oraz że Jaś skłamał już dokładnie raz.

Bazą wiedzy nazwiemy komplet informacji, jakie Małgosia może wywnioskować po zadaniu Jasiowi pewnej liczby pytań. W dotychczasowych rozważaniach bazy wiedzy wyrażaliśmy słownie, ale teraz spróbujemy sformalizować nieco ich opis. Możemy, na przykład, zauważyć, że po zadaniu dowolnej liczby pytań baza wiedzy Małgosi będzie n-elementowym ciągiem (a_i), określającym, dla każdego i, liczbę dotychczasowych kłamstw Jasia, o ile $x = i$. Zakładając, że Jaś nie oszukuje w swoich odpowiedziach (bardziej niż mu na to pozwalają zasady), to $a_i \geqslant 2$ oznacza po prostu, że $x \neq i$, czyli możemy ograniczyć naszą uwagę do ciągów nad $\{0, 1, 2\}$. Jest to stosunkowo miła informacja; wynika z niej, że możliwych baz wiedzy jest skończenie wiele (a dokładniej 3^n).

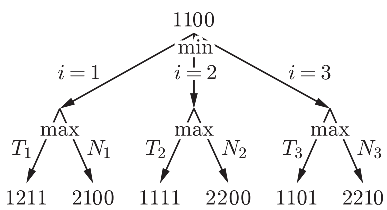
Więcej o programowaniu dynamicznym można poczytać w dowolnej książce o algorytmice, np.: T. Cormen, C. Leiserson, R. Rivest, C. Stein, *Wprowadzenie do algorytmów*.

Umiemy więc już w systematyczny sposób interpretować ciągi odpowiedzi Jasia, ale… wciąż nie wiemy właściwie, jak należy zadawać pytania! W naszej strategii optymalnej wykorzystamy więc *programowanie dynamiczne*, tzn. spróbujemy dla każdej możliwej bazy wiedzy obliczyć liczbę pytań, za pomocą których można, mając wszystkie informacje z tej bazy, odgadnąć faktyczną wartość x. Znów interesować nas będzie *najgorszy możliwy przypadek*, czyli najbardziej złośliwa wobec Małgosi kombinacja wartości x oraz momentu, w którym Jaś skłamie (jeżeli w ogóle zdecyduje się to uczynić). Cały proces zaczynamy od bazy odpowiadającej ciągowi zerowemu, a kończymy na ciągach z co najwyżej jedną wartością różną od 2.

Zauważmy istotną różnicę między tym podejściem a optymalną strategią dla prostej gry w zgadywanie. Poprzednio Małgosia miała sztywny schemat zadawania pytań, który po prostu konsekwentnie realizowała. Tym razem jeszcze przed samą rozgrywką Małgosia zasymuluje sobie rozmaite możliwości na kartce (lub na komputerze), a dopiero potem zastosuje najlepszą strategię, jaką znalazła.

Tak naprawdę wykorzystaliśmy tzw. rekurencję ze spamiętywaniem, a nie programowanie dynamiczne, ale w gruncie rzeczy techniki te są różnymi zapisami tej samej metody.

Ta procedura może się zapętlić, ale tylko wtedy, gdy $T_i = A$ lub $N_i = A$. Niemniej jednak w takim przypadku i-tego pytania możemy po prostu nie rozważać.



Schemat wyznaczania wyniku dla $A = 1100$.

Jak więc obliczyć wynik dla danego ciągu $A = a_1, \ldots, a_n$ reprezentującego pewną bazę wiedzy, tj. $w(A)$? Przeglądamy wszystkie możliwe pytania („Czy $x \leqslant i$?” dla $i = 1, 2, \ldots, n-1$). Dla i-tego pytania rozważamy bazy wiedzy

$$T_i = a_1, \ldots, a_i, \min(a_{i+1}+1, 2), \ldots, \min(a_n+1, 2)$$

oraz

$$N_i = \min(a_1+1, 2), \ldots, \min(a_i+1, 2), a_{i+1}, \ldots, a_n,$$

które otrzymujemy, jeżeli odpowiedzią na to pytanie jest odpowiednio „Tak” lub „Nie”. Dla każdej z wyznaczonych baz wyliczamy rekurencyjnie wartość w (lub wykorzystujemy gotową wartość, jeżeli została ona już obliczona we wcześniejszych krokach algorytmu). Ponieważ dążymy do wyboru najlepszego możliwego pytania, a dla każdego pytania zakładamy najbardziej pesymistyczny wariant odpowiedzi Jasia, to:

$$w(A) = \min_{1 \leqslant i \leqslant n-1} \{\max(w(T_i), w(N_i))\}.$$

Koszt czasowy obliczenia tego wszystkiego to $O(3^n \cdot n)$.

Zauważmy, że na podstawie wartości w stosunkowo łatwo jest skonstruować optymalną strategię dla Małgosi. Faktycznie, jeżeli po zadaniu pewnej liczby pytań bazą wiedzy jest A, to Małgosia jako następne pytanie powinna wybrać „Czy $x \leqslant i$?”, gdzie i spełnia: $\max(w(T_i), w(N_i)) = w(A)$. Nieco zaskakujące może być to, że również Jaś może uczynić pożytek z tych wartości w swojej strategii, jeżeli – tak jak poprzednio – postanowi dokonać niewykrywalnego oszustwa, tzn. wybrać wartość x dopiero w trakcie gry. W tym celu na pytanie Małgosi „Czy $x \leqslant i$?” Jaś powinien odpowiedzieć tak, aby przejść do tej z baz T_i oraz N_i, dla której wynik jest większy, o ile ta baza nie jest sprzeczna, czyli nie jest ciągiem złożonym z samych dwójek.

Skonstruowaliśmy więc poszukiwane strategie; problem tkwi w tym, że złożoność czasowa ich wyznaczenia jest wykładnicza, co już dla niewielkich n może być dla naszych graczy bardzo kłopotliwe, nawet jeśli użyją komputera. Okazuje się jednak, że złożoność czasową tego procesu można istotnie zmniejszyć, redukując rozmiar zbioru baz wiedzy. Każda baza wiedzy jest sumą pewnej liczby ciągów typu $1\ldots10\ldots0$ oraz typu $0\ldots01\ldots1$. Jak można łatwo sprawdzić (do czego zachęcamy Czytelnika), suma takich ciągów musi być postaci:

I. $2\ldots21\ldots10\ldots01\ldots12\ldots2$ lub
II. $2\ldots21\ldots12\ldots21\ldots12\ldots2$.

W przypadku I z punktu widzenia strategii istotne są jedynie liczby kolejnych jedynek, zer i jedynek, więc mamy $O(n^3)$ istotnie różnych baz tej postaci. Z kolei strategia gry dla bazy postaci II jest praktycznie taka sama jak dla bazy złożonej z samych jedynek w takiej samej ich liczbie, czyli jest to po prostu wyszukiwanie binarne (brak zer – Jaś nie może już więcej kłamać). Zredukowaliśmy zatem moc zbioru baz wiedzy do $O(n^3)$, a złożoność algorytmu do wielomianowej: $O(n^4)$. Okazuje się, że także i ten wynik można jeszcze poprawić (tym razem nie następuje już redukcja liczby możliwych baz wiedzy), osiągając złożoność $O(n^3 \log n)$. W tym celu trzeba zauważyć, że wynik dla dowolnej bazy jest rzędu $O(\log n)$ oraz że przy danej bazie A dla $i < j$ zachodzi $w(T_i) \leqslant w(T_j)$ i $w(N_i) \geqslant w(N_j)$, a następnie uczynić pożytek z tych spostrzeżeń za pomocą kilku dodatkowych tablic w algorytmie. Szczegóły tego usprawnienia pozostawiamy Czytelnikowi, podobnie jak uogólnienie opisanego rozwiązania na wersję gry, w której Jaś może skłamać co najwyżej k razy.

Czy zauważyliście, w którym momencie pokazaliśmy ograniczenie na wartość wyniku?

Na koniec przedstawiamy jeszcze jedną ciekawą zagadkę dla Czytelnika. Mamy j jajek, z których każde ma wytrzymałość $h \in \{0, \ldots, H\}$, tzn. zrzucone z wysokości h nie rozbija się, ale z wysokości $h+1$ już tak. Nie znamy wartości h, a jedynie j oraz H; chcemy wyznaczyć h za pomocą najmniejszej możliwej liczby zrzutów jajek. Jajko, które po zrzuceniu się nie rozbije, może być wykorzystane do dalszych prób. Jak zaprojektować eksperyment? Jak zmodyfikować strategię, jeżeli co najwyżej jedno jajko może być wadliwe, tzn. rozbijać się po zrzuceniu z wysokości nie większej niż h? Wreszcie jak postępować, jeżeli jest maksymalnie k różnych wadliwych jajek?