protokołem. Komunikatem może być praktycznie wszystko: list miłosny, fragment zakodowanego filmu, wirus komputerowy, sygnał uruchamiający ogrzewanie w moim domu itp. Nie wszystkie pakiety składające się na komunikat idą tą samą drogą od nadawcy do odbiorcy; węzły transmisji wybierają dla każdego z osobna drogę najtańszą. W rezultacie, pakiety mogą docierać nie po kolei, ostateczny montaż odbywa się w komputerze odbiorcy. To, że korzystając z Internetu, mamy wrażenie pełnej ciągłości i płynności dźwięku i obrazu, świadczy o mocy sprzętowych środków informatyki (hardware) i skuteczności oprogramowania (software) tworzących światową sieć komputerową (WWW).

Protokoły internetu są jednakowe dla wszystkich. Stosują je uczniowie i hobbyści, uczeni i maklerzy, terroryści i policjanci. Umożliwia to globalne funkcjonowanie Internetu, lecz także, niestety, ułatwia stale rosnącą przestępczość w sieci. Paradoksalnie, gdyby sieć nie rozwinęła się tak szybko, a raczej gdyby jej rozwój nie nastąpił tak wcześnie, sytuacja mogłaby być inna.

Od drugiej wojny światowej kryptologia (wiedza o szyfrowaniu) zmieniła swój charakter. Od czasu, gdy M. Rejewski z kolegami czysto matematycznym wnioskowaniem (opartym na teorii grup) złamali szyfr Enigmy i zbudowali mechaniczne szablony (dla niepoznaki i żartu nazwane bombami) do masowego dekodowania niemieckich depesz wojskowych, amerykańscy kryptolodzy też przez czysto matematyczną analizę złamali japońskie szyfry i wreszcie A. Turing zbudował (w 1943/44) elektroniczny komputer (Colossus, tajny do lat 80. ub. w.) do deszyfrowania niemieckich depesz kodowanych ulepszonymi wersjami Enigmy i zupełnie nowymi Lorentzami – stało się oczywiste, że kryptologia wychodzi z epoki płaszcza i szpady i staje się gałęzią matematyki i informatyki. Wzbudziła ożywione zainteresowanie analityczną teorią liczb, którą mało kto się zajmował, przywróciła blask teorii funkcji eliptycznych, od stu bez mała lat uważanej za zamkniętą księgę, zapoczątkowała prace nad konstrukcją stosunkowo tanich urządzeń do szybkiego operowania na bardzo wielocyfrowych liczbach całkowitych. Dwa wspaniałe odkrycia kryptologii: asymetryczna kryptografia z kluczem publicznym (pozwalająca ujawniać klucz do kodowania, zachowując w tajemnicy klucz do rozkodowania) i podpis elektroniczny (dający gwarancję autentyczności dokumentu i jego autorstwa) zmieniłyby oblicze Internetu, szalenie utrudniając życie oszustom i figlarzom, gdyby tylko były powszechnie dostępne wtedy, gdy tworzyły się protokoły sieciowe. Dodanie ich do sieci teraz wymaga sporych modyfikacji i jest zbyt kosztowne na to, żeby zyskać popularność.

Rozwiązanie zadania F 699.
W czasach Perelmana sądzono, iż piorun realizuje się średnio przy różnicy potencjału 1000 MV, dziś encyklopedie podają tylko 100 MV. Co do natężenia prądu w piorunie, dane są jeszcze bardziej płynne; podaje się, że osiąga średnio 20 kA. Daje to łącznie moc 2 miliony MW, czyli 2 miliardy kW, z czego należy do rachunku wpisać połowę, gdyż potencjał podczas wyładowania spada do zera. Wyładowanie trwa średnio tylko 10^{-4} s, co stanowi $28 \cdot 10^{-9}$ godziny. Łącznie Zeus musi zatem zapłacić za około 28 kWh, czyli 7 euro. Nic więc dziwnego, że – nawet jeśli ma nie najwyższą emeryturę – pozwala sobie czasami na długotrwałe burze.

Rozwiązanie zadania M 1180.
Zauważmy, że

$$(ac+bd)^2+1 = \\ = (ac+bd)^2+(ad-bc)^2 = \\ = (a^2+b^2)(c^2+d^2)\,.$$

Jeśli więc $p>0$ jest wspólnym dzielnikiem liczb $ac+bd$ i a^2+b^2, to p jest dzielnikiem liczby 1, czyli $p=1$. To oznacza, że liczby $ac+bd$ i a^2+b^2 są względnie pierwsze.

Najprostszy algorytm sortowania

Rozważmy problem sortowania n-elementowej tablicy `a[1..n]`. Jaki algorytm pozwoli nam zrobić to najprościej? Nie chodzi tu wcale o szybkość, ale o to, by dało się go zaprogramować w kilku linijkach, bez żadnego ryzyka popełnienia błędu, nawet jeśli nie jesteśmy akurat w najlepszej dyspozycji umysłowej (zaraz, zaraz, miałem porównać `a[i]` z `a[i+1]` czy z `a[i-1]`? czy wewnętrzna pętla miała się obracać, dopóki `i<j`, czy `i<=j`? oj...).

No więc, który algorytm jest najprostszy? Chyba żaden z działających w czasie $O(n \log n)$ się nie nadaje. To może sortowanie przez wstawianie albo bąbelkowe? Nic z tego. Najprościej jest tak:

```
sort(a[1..n]) {
    for i:=1 to n do
    for j:=1 to n do
      if a[i] < a[j] then zamień(a[i], a[j]);
}
```

Szybki test: czy ten algorytm sortuje tablicę `a` rosnąco czy malejąco? Nie jest to jasne na pierwszy rzut oka, podobnie jak to, że tablica w ogóle zostanie jakkolwiek posortowana!

Po chwili zastanowienia stwierdzamy, że powyższy algorytm to jedno z wcieleń sortowania przez wstawianie (Insertion Sort). Na początku i-tego obrotu zewnętrznej pętli fragment `a[1..i-1]` jest posortowany (rosnąco), a wewnętrzna pętla wstawia w odpowiednie miejsce element `a[i]`, używając i-tej pozycji w tablicy jako „bufora” na przesuwane elementy początkowego fragmentu. Czytelnik łatwo sprawdzi, że wewnętrzna pętla mogłaby zakończyć działanie dla `j=i`, ale jej przedłużenie aż do `j=n` nic nie popsuje.

Nie jest to specjalnie efektywne, ale za to jakie eleganckie! Prościej już się chyba nie da...

Michał ADAMASZEK