

Samotne zwierzę na Arce Noego

Tomasz JANISZEWSKI

Noe wybudował arkę. Wybrał pary zwierząt spośród wszystkich gatunków i każdemu gatunkowi przyporządkował liczbę naturalną. Przed zamknięciem arki nakazał jednemu z synów spisać numery gatunków wszystkich zwierząt. Niestety okazało się, że brakuje jednego zwierzęcia. Którego?

Powyższe pytanie odpowiada następującemu zadaniu: Mając daną tablicę $A[1, n]$ zawierającą n liczb naturalnych, znajdź liczbę, która występuje dokładnie raz, wiedząc, że pozostałe liczby występują po dwa razy.

Na pierwszy rzut oka można to zadanie rozwiązać w czasie $O(n \log n)$. Możemy bowiem posortować tablicę i przejść po niej, szukając pierwszego elementu, którego poprzednik i następnik będą od niego różne.

```

function SZUKAJ(A)
  posortuj A
  if A[1] ≠ A[2] then
    return A[1]
  for i := 2 to n - 1 do
    if A[i] ≠ A[i - 1] and A[i] ≠ A[i + 1] then
      return A[i]
  return A[n]

```

Okazuje się, że można to zadanie rozwiązać w czasie liniowym. W tym celu należy skorzystać ze zbioru. Zbiór (*set*) to abstrakcyjna struktura zoptymalizowana pod kątem sprawdzania, czy dany element do niej należy, czy też nie. Podstawowe funkcje zbioru to dodawanie i usuwanie elementów, jak również sprawdzanie, czy element należy do zbioru. Zbiór może być realizowany za pomocą różnych struktur danych, takich jak tablica haszująca, drzewo czy nawet zwykła tablica. Każda implementacja ma swoje wady i zalety. Dla uproszczenia przyjmijmy, że operacje dodawania, usuwania i sprawdzania, czy element należy do zbioru, mają stały koszt czasowy $O(1)$. Jeśli dysponujemy zbiorem, to zadanie możemy rozwiązać w następujący sposób. Przechodząc po tablicy, sprawdzamy, czy dany element znajduje się w zbiorze; jeśli nie, to go dodajemy, a jeśli tak – usuwamy. Na końcu jedynym elementem zbioru powinien być szukany gatunek.

```

function SZUKAJ(A)
  S := ∅
  for i := 1 to n do
    if A[i] ∈ S then
      S := S \ {A[i]}
    else
      S := S ∪ {A[i]}
  return S

```

Rozwiązanie to działa w czasie liniowym $O(n)$, ale wymaga dodatkowej pamięci, która w pesymistycznym przypadku również będzie liniowo zależna od tablicy wejściowej. Rodzi się pytanie: czy można to zrobić równie szybko i zużyć mniej pamięci? Okazuje się, że tak – wystarczy skorzystać z operacji XOR i z tego, jak w komputerze reprezentowane są liczby.

Jest wiele sposobów reprezentacji liczb. Nas interesują wyłącznie liczby naturalne, a to znacznie ułatwia zadanie, gdyż ich reprezentacja nie odbiega znacząco od tego, do czego jesteśmy przyzwyczajeni na co dzień. Jedyną różnicą to podstawa systemu pozycyjnego – komputer w odróżnieniu od ludzi korzysta z systemu dwójkowego, a nie dziesiętnego. Dla przykładu 49 w systemie dziesiętnym to 110001 w systemie dwójkowym ($49 = 2^5 + 2^4 + 0 + 0 + 0 + 2^0$).

XOR (z angielskiego *eXclusive OR*, oznaczane \oplus) to bitowa wersja polskiego „albo”, czyli alternatywa wykluczająca: zwraca prawdę, kiedy jeden z argumentów jest prawdziwy, a drugi fałszywy. W binarnej reprezentacji liczb

Zakładamy, że liczby w tablicy są „małe” (np. mniejsze niż 2^{32}) i ich porównywanie oraz operacje bitowe (takie jak XOR, o czym za chwilę) mają stały koszt czasowy $O(1)$.



Rozwiązanie zadania F 1031.

Ze stałych wymienionych w definicji Układu SI należy zbudować wyrażenia o wymiarze 1 kg oraz 1 Ω. Kilogram występuje w wymiarze stałej h (zawarty jest w definicji dżula). Wymiar stałej k także zawiera 1 kg, ale poza tym występuje też 1 K, którego nie ma w wymiarze żadnej innej stałej. Będziemy potrzebowali poza h także stałych o wymiarach zawierających 1 m i 1 s. Zbadajmy wymiar iloczynu:

$$h^\alpha \cdot c^\beta \Delta\nu_{Cs}^\gamma$$

i dobierzmy wykładniki tak, żeby wymiar iloczynu zawierał wyłącznie kilogram w pierwszej potędze. Wymiar badanego iloczynu to:

$$J^\alpha s^\alpha \cdot \frac{m^\beta}{s^\beta} \cdot \frac{1}{s^\gamma} = \frac{kg^\alpha m^{2\alpha} s^\alpha m^\beta}{s^{2\alpha\beta} s^\gamma}$$

Iloczyn ma wymiar kilograma, jeśli:

$$\alpha = 1, 2\alpha + \beta = 0 \text{ oraz } \alpha + \beta + \gamma = 0.$$

Otrzymujemy: $\alpha = 1, \beta = -2, \gamma = 1$. Po

podstawieniu wartości liczbowych:

$$h \cdot c^{-2} \Delta\nu_{Cs} = 6,777265 \dots \cdot 10^{-41} \text{ kg,}$$

a więc 1 kg =

$$1,475521 \dots \cdot 10^{40} h \Delta\nu_{Cs} / c^2. \text{ W podobny}$$

sposób sprawdzamy, że

$$h/e^2 = 25812,807459 \dots \Omega, \text{ czyli}$$

$$1 \Omega = 3,87404586 \dots \cdot 10^{-5} h/e^2.$$

Tablica prawdy XOR:

A	B	A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

XOR sprowadza się do sumy bez przeniesienia:

$$\begin{array}{r} 49_{10} \ 110001 \\ \oplus \ 3_{10} \ 000011 \\ \hline 50_{10} \ 110010 \end{array}$$

Na przykład:

$$\begin{array}{r} 10_{10} \ 01010 \\ 21_{10} \ 10101 \\ 19_{10} \ 10011 \\ 19_{10} \ 10011 \\ \oplus \ 10_{10} \ 01010 \\ \hline 21_{10} \ 10101 \end{array}$$

Widzimy, że operacja XOR jest przemienne ($A \oplus B = B \oplus A$) i łączna ($(A \oplus B) \oplus C = A \oplus (B \oplus C)$), a zatem wynik nie zależy od kolejności elementów. Oznacza to, że kiedy zastosujemy XOR do wszystkich elementów tablicy, to rezultat będzie taki sam, jak gdyby wszystkie pary takich samych elementów stały obok siebie. Ale przecież dla każdej liczby A zachodzi $A \oplus A = 0$. Wynikiem tej operacji będzie liczba, która występuje tylko raz! W ten sposób dostajemy rozwiązanie działające w czasie liniowym ($O(n)$), które potrzebuje stałej ilości pamięci.

```
function SZUKAJ(A)
  x := 0
  for i := 1 to n do
    x := x ⊕ A[i]
  return x
```



O błędach pozycji gwiazd i prędkościach cząsteczek gazu

Alex SCHWARZENBERG-CZERNY*

* Centrum Astronomiczne im. Mikołaja Kopernika w Warszawie

John Herschel (1792–1871) dokonał wielu pomiarów pozycji gwiazd na niebie, w tym po raz pierwszy w historii na tak dużą skalę na półkuli południowej, prowadząc obserwacje z przyłądka Dobrej Nadziei. Przy okazji zainteresował się błędami swoich pomiarów i tym, jaki może być ich rozkład. Na podstawie sposobu pomiaru i charakteru obrazu gwiazdy w teleskopie uznał, że

- (a) prawdopodobieństwo błędu zależy od jego wielkości r , ale nie od kierunku na niebie, oraz że
- (b) błędy we współrzędnych x i y są niezależne.

Dodatkowo prawdopodobieństwo błędu winno zmieniać się płynnie z pozycją gwiazdy, to znaczy być ciągią funkcją współrzędnych. Na podstawie obserwacji z podpunktu (a) prawdopodobieństwo dP , że pomiar wypadnie w kawałku nieba o powierzchni dS w odległości r od prawdziwej pozycji, możemy zapisać jako $dP = g(r) dS$. Natomiast na podstawie (a) i (b) prawdopodobieństwo, że pomiar wypadnie w kawałku nieba o powierzchni dS , będzie równe $dP = f(x)f(y)dS$ dla pewnej funkcji ciągłej f . Mamy zatem

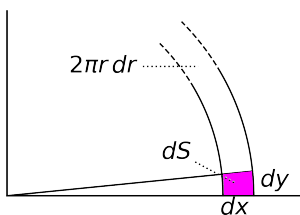
$$(1) \quad g(r) dS = dP = f(x)f(y) dS, \quad \text{co oznacza, że} \quad \ln g(r) = \ln f(x) + \ln f(y).$$

Jeśli za początek układu współrzędnych wybierzemy prawdziwą pozycję gwiazdy, to jak nauczał Pitagoras, $r^2 = x^2 + y^2$. Gdybyśmy funkcje $\ln f(x)$, $\ln g(r)$ mogli przedstawić w postaci szeregu („nieskończonego wielomianu”), czyli

$$\ln g(r) = -a - br - cr^2 - dr^3 - \dots,$$

to moglibyśmy zauważyć, że tylko dwa wyrazy (pierwszy i trzeci) powyższego szeregu można przedstawić w postaci sumy funkcji x i y , a zatem $\ln g(r) = -a - cr^2 = -a - c(x^2 + y^2)$. Rozwijanie w szereg to chleb powszedni wielu naukowców, których powinno ucieszyć to obrazowe uzasadnienie. Dla Czytelników, którym wydaje się ono niewystarczająco precyzyjne, przedstawimy też inne wyprowadzenie. Niech $\alpha(t) = \ln g(\sqrt{t})$, $\beta(x) = \ln f(x)$. Wstawiając do (1)

Ogólnie prawdopodobieństwo $P(A \cap B)$ zajścia na raz zdarzeń A i B , które mogą być zależne, wyraża się iloczynami $P(A \cap B) = P(A)P(B|A) = P(B)P(A|B)$, gdzie $P(B|A)$ to prawdopodobieństwo zajścia B , gdy wiadomo, że zachodzi A , i odwrotnie dla $P(A|B)$. W skrajnym wypadku, gdy muszą one zachodzić na raz, $P(B|A) = P(A|B) = 1$, tj. jedno jest pewną konsekwencją drugiego i odwrotnie. Natomiast, gdy są one niezależne, to można opuścić drugi argument i $P(B|A) = P(B)$, $P(A|B) = P(A)$.



Obszar wokół rzeczywistej pozycji gwiazdy