

Podobnie jak wyznaczanie liczb pierwszych, problem znajdowania największego wspólnego dzielnika (NWD) dwóch liczb naturalnych ma klasyczne i zapewne wszystkim Czytelnikom znane rozwiązanie. Mowa oczywiście o algorytmie Euklidesa, jednym z najstarszych do dziś używanych algorytmów. Jest to rozwiązanie bardzo proste w implementacji, a w wersji z dzieleniem – efektywne: pozwala wyznaczyć NWD dwóch liczb nie większych niż n w czasie $O(\log n)$. W pewnych przypadkach istnieją jednak rozwiązania asymptotycznie szybsze.

Załóżmy, że chcemy wyznaczać największy wspólny dzielnik dla wielu par stosunkowo niewielkich liczb. A konkretnie – dla q par liczb całkowitych dodatnich nieprzekraczających n . Stosując algorytm Euklidesa, otrzymujemy oczywiście algorytm o złożoności czasowej $O(q \log n)$. Możemy także spamiętać w dużej tablicy odpowiedzi na wszystkie możliwe n^2 zapytań. Wówczas przy pierwszym podejściu możemy otrzymać złożoność $O(n^2 \log n + q)$, gdy dla każdej możliwej pary zastosujemy niezależnie algorytm Euklidesa. Bez problemów takie rozwiązanie możemy przyspieszyć do $O(n^2 + q)$, ponieważ pojedynczy krok algorytmu Euklidesa sprowadza wyznaczanie NWD dla pary (i, j) do problemu obliczenia NWD dla pewnej pary leksykograficznie mniejszej. Możemy zatem wypełniać tablicę w odpowiedniej kolejności i w każdym kroku przepisywać wynik ze wskazanej komórki. W praktyce lepiej w tej sytuacji korzystać z algorytmu Euklidesa w wersji z odejmowaniem, gdyż procesory wykonują je istotnie szybciej niż wyznaczanie reszty z dzielenia. Otrzymujemy w ten sposób następujący algorytm:

```

Algorytm tablicuj-nwd( $n$ )
  for  $i := 1$  to  $n$  do
     $nwd[i, i] := i$ ;
  for  $i := 1$  to  $n$  do
    for  $j := 1$  to  $i - 1$  do
       $nwd[j, i] := nwd[i, j] := nwd[j, i - j]$ ;
    
```

Czas obliczeń wstępnych rzędu $O(n^2)$ jest jednak mało satysfakcjonujący – dopiero przy $\Omega(n^2 / \log n)$ zapytaniach podejście korzystające z algorytmu Euklidesa przestaje osiągać lepsze wyniki.

Spróbujmy odnieść pewne korzyści już z obliczeń wstępnych rzędu $O(n)$. Jak przekonaliśmy się w poprzednim artykule, w tym czasie można wyznaczyć wszystkie liczby pierwsze nieprzekraczające n . Zaprezentowany tam algorytm jest jednak znacznie silniejszy – dla każdej liczby k oblicza jej najmniejszy dzielnik pierwszy (oznaczany tu przez $ndp[k]$) oraz największą potęgę tego $ndp[k]$ dzielącą k . Za ich pomocą można łatwo skonstruować algorytm znajdowania NWD, działający w czasie proporcjonalnym do liczby różnych dzielników pierwszych argumentów. Pesymistycznie wielkość ta wynosi $O(\frac{\log n}{\log \log n})$, więc zapytania obsługiwane są niewiele szybciej niż przez algorytm Euklidesa. Okazuje się jednak, że przy liniowych obliczeniach wstępnych można osiągnąć stały czas zapytania!

Na wstępie zauważmy, że jeśli jeden z argumentów jest liczbą pierwszą, obliczenie NWD jest bardzo łatwe. Również łatwo możemy wyznaczać NWD liczb mniejszych niż \sqrt{n} : jeśli mamy do dyspozycji czas $O(n)$ na obliczenia wstępne, można spamiętać wszystkie wartości NWD, wywołując *tablicuj-nwd*($\lfloor \sqrt{n} \rfloor$). Jak się wkrótce przekonamy, dowolne zapytanie o NWD można sprowadzić do stałej liczby zapytań, w których każdy z argumentów jest liczbą pierwszą lub nie przekracza \sqrt{n} . Kluczowe jest tu pojęcie *rozkładu specjalnego*.

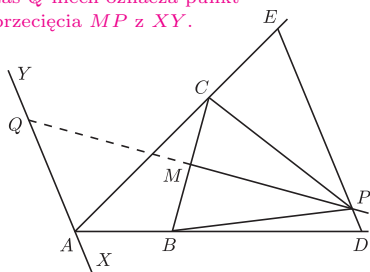
Definicja. *Rozkładem specjalnym* dodatniej liczby całkowitej k nazwiemy trójkę liczb całkowitych (k_1, k_2, k_3) , dla której $k_1 k_2 k_3 = k$ oraz dla każdego i :

$$k_i \leq \sqrt{k} \quad \text{lub} \quad k_i \text{ jest liczbą pierwszą.}$$



Rozwiązanie zadania M 1372.

Poprowadźmy prostą XY przechodzącą przez punkt A i równoległą do DE . Niech M będzie środkiem BC , zaś Q niech oznacza punkt przecięcia MP z XY .



Skoro punkt M jest środkiem BC , jego odległość od prostej DE to średnia arytmetyczna odległości punktów B i C od DE . Jest ona równa średniej arytmetycznej odległości tych punktów od XY , ponieważ $AB/BD = CE/CA$ i $XY \parallel DE$. Zatem M jest równo odległy od DE i XY , skąd $MQ = MP$ oraz $\sphericalangle BQC = \sphericalangle BPC$.

Niech Q' będzie takim punktem na półprostej MQ , że $\sphericalangle BQ'C = \sphericalangle BAC$. Z podobieństwa trójkątów równoramiennych DAE i $BQ'C$ mamy $\sphericalangle ADE = \sphericalangle Q'CB$. Zatem skoro na czworokącie $BAQ'C$ można opisać okrąg, to

$$\begin{aligned} \sphericalangle BAQ' &= 180^\circ - \sphericalangle Q'CB = \\ &= 180^\circ - \sphericalangle ADE = \\ &= 180^\circ - \sphericalangle DAX = \sphericalangle BAQ. \end{aligned}$$

Wobec tego $Q = Q'$, co daje tezę.

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

Przykładowo, jednym z rozkładów specjalnych liczby 156 jest (2, 6, 13), liczby 175 jest (5, 5, 7), a liczby 1 jest (1, 1, 1).

Następujący lemat stanowi podstawę indukcyjnego dowodu faktu, że każda dodatnia liczba całkowita ma rozkład specjalny. Ten dowód można natychmiast przekształcić w algorytm wyznaczający rozkłady specjalne wszystkich liczb od 1 do n w czasie liniowym na podstawie tablicy ndp .

Lemat. Niech $l > 1$ będzie liczbą całkowitą, $p = ndp[l]$ oraz $k = l/p$. Niech (k_1, k_2, k_3) będzie rozkładem specjalnym k , dla którego $k_1 \leq k_2 \leq k_3$. Wówczas $(k_1 \cdot p, k_2, k_3)$ jest rozkładem specjalnym l .

Dowód. Na początek zauważmy, że liczby k_2 i k_3 są pierwsze lub nie większe niż \sqrt{k} , więc są pierwsze lub nie większe niż \sqrt{l} . Musimy wobec tego zająć się tylko liczbą $k_1 \cdot p$. Jeśli $k_1 = 1$, to oczywiście $k_1 \cdot p = ndp[l]$ jest liczbą pierwszą. Załóżmy więc, że $k_1 > 1$. Ponieważ k_1 jest dzielnikiem l , a p jest najmniejszym (różnym od 1) dzielnikiem l , więc $p \leq k_1$. Wobec tego

$$(k_1 p)^2 = k_1^2 p^2 \leq p k_1^3 \leq p k_1 k_2 k_3 = p k = l,$$

a więc $k_1 p \leq \sqrt{l}$, co kończy dowód. \square

Pozostaje nam jeszcze wykorzystać rozkłady specjalne do efektywnego obliczania NWD. Tym razem najpierw przedstawimy algorytm, a potem zastanowimy się nad jego poprawnością. Przyjmiemy, że rozkłady specjalne zapamiętaliśmy w tablicy *rozklad*.

Algorytm $nwd(k, l)$

```

 $(x_1, x_2, x_3) := rozklad[k];$ 
 $(y_1, y_2, y_3) := rozklad[l];$ 
 $g := 1;$ 
foreach  $i, j \in \{1, 2, 3\}$  do
  if  $\max(x_i, y_j) \leq \sqrt{n}$  then
     $d := nwd[x_i, y_j];$ 
  else if  $x_i = y_j$  then  $d := x_i;$ 
  else  $d := 1;$ 
   $g := g \cdot d;$ 
   $x_i := x_i/d; y_j := y_j/d;$ 
return  $g;$ 

```

Na początek zauważmy, że w każdym obrocie pętli d jest pewnym wspólnym dzielnikiem x_i oraz y_j . Co więcej, d jest największym wspólnym dzielnikiem x_i oraz y_j . Jeśli $x_i, y_j \leq \sqrt{n}$ lub $x_i = y_j$, nie mamy co do tego wątpliwości. Bez straty ogólności niech więc $x_i < y_j$ oraz $\sqrt{n} < y_j$. Wówczas y_j jest liczbą pierwszą, a więc $nwd(x_i, y_j)$ może być równe 1 lub y_j . Drugą możliwość natychmiast wyklucza nierówność $x_i < y_j$.

Pozostaje teraz wykazać, że $g = nwd(k, l)$. Zrobimy to przez wskazanie niezmienników pętli:

- $k = x_1 \cdot x_2 \cdot x_3 \cdot g$,
- $l = y_1 \cdot y_2 \cdot y_3 \cdot g$,
- $nwd(x_i, y_j) = 1$ dla już przetworzonych par i, j .

Na ich podstawie wnioskujemy, że po ostatnim obrocie pętli g jest wspólnym dzielnikiem k i l , a liczby k/g i l/g są względnie pierwsze.

W ten sposób skonstruowaliśmy algorytm, który na q zapytań o NWD liczb z zakresu $1, \dots, n$ odpowiada w czasie $O(n + q)$. O ile tylko $q > \frac{n}{\log n}$, to ten algorytm jest najlepszy z tutaj przedstawionych. Okazuje się, że w pewien sposób można połączyć jego siły z algorytmem Euklidesa i otrzymać rozwiązanie, które nigdy nie jest asymptotycznie wolniejsze od żadnego z zaprezentowanych, a dla pewnych wartości q (np. $q = \frac{n}{\log n}$) jest niemal $\log n$ razy szybsze. Stworzenie tego działającego w czasie $O(q \cdot \max(1, \log \frac{n}{q}))$ algorytmu pozostawiamy Czytelnikowi jako zadanie.



Rozwiązanie zadania M 1374.

Zauważmy najpierw, że dla liczby całkowitej x zachodzi $3 \mid x^{2013} - x$. Stąd

$$L = 1^{2013} + 2^{2013} + \dots + (p-1)^{2013} \equiv 1 + 2 + \dots + (p-1) \pmod{3}.$$

Jeśli więc p jest postaci $3k + 1$, to wówczas

$$L \equiv (1 + 2) + 3 + \dots + (3k - 2 + 3k - 1) + 3k \equiv 0 \pmod{3},$$

zaś $p^{2013} \equiv p \equiv 1 \pmod{3}$, więc równanie nie ma rozwiązania w tym przypadku. Podobnie stwierdzamy, że dla p postaci $3k + 2$ równanie jest sprzeczne. Zatem jedyną możliwością to $p = 3$, ale łatwo sprawdzić, że wtedy równanie również jest sprzeczne.

Uwaga. Paul Erdős około 1950 roku w liście do Leo Mosera postawił hipotezę, że równanie

$$1^k + 2^k + \dots + (m-1)^k = m^k$$

nie ma rozwiązań w liczbach naturalnych m oraz $k \geq 2$. Dziś wiadomo, że jeśli równanie ma rozwiązanie dla jakiegoś $k \geq 2$, to $m > 10^{10^9}$ (zgrabny dowód i historia problemu są przedstawione w artykule: P. Moore, *A top hat for Moser's four mathematical rabbits*, Amer. Math. Monthly 118 (2011), 364–370).