Zajecia z informatyki, LO2 Opole

Algorytm Euklidesa

15.01.2022

Rozważmy następujący problem:

Wejście: Dodatnie liczby naturalne a i b.
Wyjście: Największy wspólny dzielnik a i b, który będziemy oznaczać przez $gcd(a, b)$[1].
Przykłady:

- $gcd(12, 18) = 6$
- $gcd(15, 9) = 1$
- $gcd(10, 20) = 10$

1 Algorytm naiwny

Najprostszym pomysłem jest po prostu sprawdzenie wszystkich możliwych liczb. Iterując się od największych do najmniejszych kandydatów zapewniamy, że zatrzymamy się na największym z nich.

```
long long gcd_naive(long long a, long long b) {
   for (long long i = min(a, b); i >= 1; --i) {
      if (a % i == 0 && b % i == 0) {
         return i;
      }
   }
   /* Do tego miejsca nasz algorytm nigdy nie dojdzie,
      poniewaz 1 dzieli dowolna liczbe. Z tego powodu
      nie musimy tutaj pisac "return", nawet jezeli kompilator
      nas ostrzega, ze o tym zapomnielismy.
   */
}
```

Pętla w powyższym algorytmie wykona się w najgorszym wypadku $min(a, b)$ razy. Dla liczb wejściowych rzędu 10^{18} na współczesnym laptopie taki program będzie się wykonywał około kilkunastu lat.

Zadanie: Podaj przykład liczb a i b mieszczących się w typie `int`, dla których pętla wykona się co najmniej $1\,000\,000\,000$ razy.

Odpowiedź: $(10^9, 10^9 + 1)$. Pętla sprawdzi wszystkie liczby dodatnie mniejsze od 10^9.

[1] greatest common divisor

2 Algorytm prawie szybki

Do skonstruowania szybszego algorytmu przyda nam się następujące twierdzenie:

Liczby naturalne a i b ($a \leq b$) mają dokładnie takie same wspólne dzielniki jak liczby a i $(b - a)$.

Dowód powyższego twierdzenia znajduje się na końcu dokumentu. Warto go zrozumieć!

Przykład:

wspólne dzielniki liczb 12 *i* 30 *to* $1, 2, 3, 6$*,*

więc

wspólne dzielniki liczb $12, 18$ *to też* $1, 2, 3, 6$.

Możemy ten proces kontynuować:

wspólne dzielniki liczb $12, 6$ *to* $1, 2, 3, 6$
(zawsze odejmujemy mniejszą liczbę od większej)

wspólne dzielniki liczb $6, 6$ *to* $1, 2, 3, 6$

wspólne dzielniki liczb $6, 0$ *to* $1, 2, 3, 6$

wspólne dzielniki liczb $6, 0$ *to* $1, 2, 3, 6$

wspólne dzielniki liczb $6, 0$ *to* $1, 2, 3, 6$

$\ldots$

Powyższe rozumowanie to właśnie prostsza wersja słynnego algorytmu Euklidesa. Żeby obliczyć $gcd(a, b)$, wystarczy, że obliczyby $gcd(a, b - a)$, ponieważ obie pary liczb mają dokładnie takie same wspólne dzielniki.

Rozwiążmy za pomocą algorytmu Euklidesa problem dla danych wejściowych $a = 153, b = 93$.

gcd(153, 93*) jest równe gcd(*93, 60*)*

gcd(93, 60*) jest równe gcd(*60, 33*)*

gcd(60, 33*) jest równe gcd(*33, 27*)*

gcd(33, 27*) jest równe gcd(*27, 6*)*

gcd(27, 6*) jest równe gcd(*21, 6*)*

gcd(21, 6*) jest równe gcd(*15, 6*)*

gcd(15, 6*) jest równe gcd(*9, 6*)*

gcd(9, 6*) jest równe gcd(*6, 3*)*

gcd(6, 3*) jest równe gcd(*3, 3*)*

gcd(3, 3*) jest równe gcd(*3, 0*)*

$gcd(3, 0)$ to oczywiście 3, więc $gcd(153, 93) = 3$.

```
long long gcd(long long a, long long b) {
  if (a > b) {       // jezeli a jest wieksze od b,
    swap(a, b);      // to zamien liczby miejscami
  }

  while (a != 0) {   // dopoki mniejsza liczba jest wieksza od 0
    b = b - a;       // odejmujemy od wiekszej liczby mniejsza liczbe
    if (a > b) {
      swap(a, b);    // i ew. zamieniamy je miejscami,
                     // zeby a nie bylo wieksze
    }
  }
  return b;
}
```

Niestety, powyższy algorytm wciąż jest bardzo wolny dla niektórych liczb wejściowych.

Zadanie: Podaj przykład liczb a i b mieszczących się w typie `int`, dla których pętla wykona się co najmniej $1\,000\,000\,000$ razy.

Odpowiedź: $(10^9, 1)$, będziemy odejmować jedynkę 10^9 razy.

3 Algorytm szybki

Żeby przyspieszyć powyższy algorytm, wystarczy zauważyć, że w bardzo łatwy sposób możemy pominąć wiele iteracji pętli. W poprzednim przykładzie obliczania $gcd(153, 93)$ mieliśmy w pewnym momencie parę liczb $27, 6$. Następnie odejmowaliśmy szóstkę od pierwszej liczby dopóki ta była większa niż 6:

$$(27, 6) \to (21, 6) \to (15, 6) \to (9, 6) \to (3, 6)$$

Moglibyśmy te cztery kroki pominąć i od razu obliczyć, że 27 zredukujemy do 3.
Zadanie: W jaki sposób?

Odpowiedź: Reszta z dzielenia! $27 \mod 6 = 3$

```
1  long long gcd(long long a, long long b) {
2    if (a > b) {
3      swap(a, b);
4    }
5    while (a != 0) {
6      b = b % a;      // <-- wystarczylo zamienic jeden znak!
7      if (a > b) {
8        swap(a, b);
9      }
10   }
11   return b;
12 }
13 // Zadanie: Okazuje sie, ze jezeli usuniemy z powyzszego kodu
14 //          linijki 2, 3, 4, 8 i 10 (ale nie 9), to algorytm wciaz
15 //          bedzie poprawny... Dlaczego?
```

Taki algorytm jest już bardzo szybki. Dla dowolnych liczb wejściowych rzędu 10^{18} pętla wykona się maksymalnie kilkadziesiąt razy, co na współczesnym laptopie zajmie kilkaset nanosekund (nanosekunda to jedna miliardowa sekundy).

Skąd pewność, że pętla tak szybko się skończy? Zauważmy, że po wykonaniu operacji `b = b % a;` liczba b zmaleje co namniej dwukrotnie, co można łatwo dowieść (pamiętajmy, że b jest większe lub równe a):

Dowód: Rozważmy dwie możliwości – a jest większe niż połowa b lub nie jest.

1. $a > \frac{b}{2}$. Wtedy `b % a` jest równe `b - a`, ponieważ a "mieści" się w b dokładnie raz. `b - a` jest z kolei mniejsze niż $\frac{b}{2}$, co wynika z założenia, że a jest większe od połowy b.
   Przykład: `30 % 16 = 30 - 16 = 14`.

2. $a \leq \frac{b}{2}$. Wtedy oczywistym jest, że `b % a` nie jest większe od połowy b, ponieważ `b % a` musi być mniejsze od a (w końcu to reszta z dzielenia).
   Przykład: `30 % 11 = 8`.

Ponieważ w każdej iteracji większa liczba maleje dwukrotnie, pętla nie może się wykonać więcej niż $\log_2(a) + log_2(b)$ razy. Dla przykładu, $log_2(10^{18}) \approx 60$.

4 Dowód twierdzenia o wspólnych dzielnikach

Przypomnijmy treść twierdzenia.

Liczby naturalne a i b ($a \leq b$) mają dokładnie takie same wspólne dzielniki jak liczby a i $(b-a)$.

Dowód będzie się składał z dwóch części. Najpierw pokażemy, że jeżeli a i b mają jakiś wspólny dzielnik, to jest on też wspólnym dzielnikiem liczb a i $(b-a)$. Następnie rozważymy twierdzenie odwrotne: jeżeli a i $(b-a)$ mają jakiś wspólny dzielnik, to na pewno jest to też wspólny dzielnik pary a i b. *Upewnij się, że rozumiesz, że obie części dowodu sa konieczne.*

Wprowadźmy standardowy zapis $d|n$, oznaczający, że liczba d dzieli liczbę n. Na przykład prawdą jest, że $3|6$, ale nie jest prawdą $3|7$.

4.1 $\rightarrow$

Chcemy udowodnić

jeżeli a i b mają jakiś wspólny dzielnik,
to jest on też wspólnym dzielnikiem liczb a i $(b-a)$.

co jest równoważne

jeżeli dla pewnego d zachodzi $d|a$ i $d|b$, to prawdą jest $d|(b-a)$.

Skoro d dzieli a, to a możemy zapisać jako

$$a = d \cdot a'.$$

gdzie a' to pewna liczba całkowita. Na przykład dla liczb $a = 15$ i $d = 3$ możemy zapisać $a = d \cdot 5$. Podobnie b możemy zapisać jako

$$b = d \cdot b'.$$

Z tego wynika, że

$$b - a = d \cdot a' - d \cdot b' = d(a' - b'),$$

co kończy dowód, ponieważ pokazaliśmy, że $(b-a)$ jest równe d razy jakaś liczba całkowita, czyli jest podzielne przez d.

4.2 $\leftarrow$

Chcemy udowodnić

jeżeli a i $(b-a)$ mają jakiś wspólny dzielnik,
to jest on też wspólnym dzielnikiem liczb a i b.

co jest równoważne

jeżeli dla pewnego d zachodzi $d|a$ i $d|(b-a)$, to prawdą jest $d|b$.

Ponownie możemy zapisać a jako

$$a = d \cdot a',$$

a $(b-a)$ jako

$$(b - a) = d \cdot x'.$$

Z tego wynika, że

$$b = (b - a) + a = d \cdot x' + d \cdot a' = d(x' + a').$$

Pokazaliśmy, że b jest równe d razy jakaś liczba całkowita, a zatem jest podzielne przez d.