

Lista Nr 03, laboratorium

(wyk. 22,23.X.08r)

1. Algorytm square-and-multiply: Dla danej liczby rzeczywistej $x \in \mathbb{R}$, $x > 0$ i liczby naturalnej $n \in \mathbb{N}$ mamy obliczyć x^n . Dwie propozycje w jaki sposób można to zrobić:

Dane: n, x
Wynik: $w = x^n$

```

1  $p \leftarrow x$ ;
2  $w \leftarrow 1.0$ ;
3 while true do
4   if  $2 \nmid n$  then // sprawdzam czy  $n$  jest nieparzyste
5      $w \leftarrow w \cdot p$ ;
6   end
7   // obliczam część całkowitą z dzielenia  $n$  przez 2
8    $n \leftarrow n \text{ div } 2$ ;
9   if  $n = 0$  then break; // koniec obliczeń
10   $p \leftarrow p \cdot p$ ;
11 end
```

Algorytm 1: „Square-and-multiply”

Dane: n, x
Wynik: $w = x^n$

```

1  $w \leftarrow x$ ;
2 for  $i \leftarrow 2$  to  $n$  do
3    $w \leftarrow w \cdot x$ ;
4 end
```

Algorytm 2: „Zwykłe” potęgowanie

Zadania:

- (a) Czy w wyniku działania obu algorytmów otrzymamy tę samą liczbę?
 - (b) Narysować *schemat blokowy* obu algorytmów.
 - (c) Napisać („źródło” w C) i uruchomić programy realizujące oba algorytmy.
 - (d) Który¹ z algorytmów wygra konkurs na tytuł „Mister Algorytmów Listy Nr 03” według kryterium podanego na wykładzie.
2. Rozszerzony algorytm Euklidesa: Dla danych liczb naturalnych $a, b \in \mathbb{N}$ wyznaczyć $p, q \in \mathbb{C}$ – całkowite o własności:

$$a \cdot p + b \cdot q = \text{NWD}(a, b).$$

3. Napisać program, który będzie sprawdzał czy podana liczba naturalna jest *liczbą pierwszą*.
4. Dla danej liczby naturalnej $n \in \mathbb{N}$ napisać program, który wyznaczy jej *rozkład na czynniki pierwsze*.
5. Sito Eratostenesa Zadanie: Dla danego $n \in \mathbb{N}$, $1 < n$ wyznaczyć wszystkie liczby pierwsze $p \in \mathbb{N}$, $p \leq n$. Opis algorytmu:

Krok i: począwszy od dwójki ustawiamy wszystkie liczby w ciąg rosnący $\{2, 3, 4, \dots, n\}$;

Krok ii: bierzemy najmniejszą liczbę z ciągu (za pierwszym razem będzie to 2, za drugim 3 itd.) i wykreślamy te pozostałe wyrazy ciągu, które są jej wielokrotnością;

Krok iii: powtarzamy **Krok ii**.

Po zakończeniu procesu skreślenia pozostaną tylko liczby pierwsze nie większe od n .

¹ Poprzestać na $n \in \{1, 2, \dots, 25\}$.

6. Hipoteza Goldbacha: *każda liczba naturalna parzysta² większa od 2 jest sumą dwóch liczb pierwszych*. Napisać program sprawdzający *hipotezę Goldbacha* dla liczb parzystych nie większych od 1000.
7. Ciąg Fibonacciego, jest określony wzorem ($n \in \mathbb{N} \cup \{0\}$):

$$F_n = \begin{cases} 0, & n = 0 \\ 1, & n = 1 \\ F_{n-1} + F_{n-2}, & n > 1 \end{cases}$$

Napisać program wyznaczający ciąg F_n dla $n \leq 25$.

8. W *pliku tekstowym* mamy zapisanych n liczb rzeczywistych. Napisać program, który po ich wczytaniu obliczy i wyświetli na ekranie *sumę* oraz ich *średnią arytmetyczną*. Nazwę pliku należy podać jako parametr przy uruchomieniu programu.

Wrocław, dnia 22/10/2008

² np. $4 = 2 + 2$