

# Wstęp do Informatyki i Programowania

## Lista nr 4 24 i 29 października

### Zadanie 1

Podaj górne ograniczenie na liczbę czynników dla rozkładu  $n$  na iloczyn liczb pierwszych. Ile maksymalnie będzie takich czynników dla liczb 64 bitowych?

### Zadanie 2

Założmy, że mamy rekurencyjną funkcję liczącą liczby Fibonacciego. Rozpisz w postaci drzewa wywołania tej funkcji dla  $n = 3$ ,  $n = 4$  i  $n = 5$ . Ile wywołań funkcji jest w tych przykładach? Postaw hipotezę, ile ich będzie dla dowolnego  $n$ .

### Zadanie 3

Napisz algorytm obliczający liczby Fibonacciego bez użycia rekurencji.

### Zadanie 4

Funkcję *silnia* na liczbach naturalnych definiujemy rekurencyjnie

$$\begin{aligned}0! &= 1 \\(n+1)! &= (n+1) \cdot n!\end{aligned}$$

1. Udowodnij, że dla każdego  $k \in \{1, \dots, n\}$  zachodzi nierówność  $(n - k + 1) \cdot k \geq n$ .
2. Wykorzystując poprzedni punkt pokaż, że  $(n!)^2 \geq n^n$ .
3. Wykorzystując poprzedni punkt oszacuj od dołu liczbę bitów potrzebnych do zapisania  $n!$  i maksymalną wartość  $n$  dla której można policzyć  $n!$  na liczbach 32 i 64 bitowych.

### Zadanie 5

Dwumian Newtona od  $n$  i  $k$  (dla  $0 \leq k \leq n$ ) definiujemy jako

$$\binom{n}{k} = \frac{n!}{(n-k)! \cdot k!}$$

Łatwo zauważyć z poprzedniego zadania, że obliczanie wartości bezpośrednio ze wzoru nie jest wskazane.

Rozważmy następujący algorytm

```
1:  $r \leftarrow 1$ 
2: for  $i$  from 1 to  $k$  do
3:    $r \leftarrow r \cdot (n - i + 1)$ 
4:    $r \leftarrow r \text{ div } i$ 
5: end for
6: return  $r$ 
```

1. Udowodnij, że powyższy kod liczy dwumian Newtona dla  $n$  i  $k$ .
2. Udowodnij, że w momencie dzielenia  $r$  przez  $i$  (linia 4),  $r$  jest wielokrotnością  $i$ .
3. Jak można poprawić ten kod aby w połowie przypadków liczył mniej.

**Zadanie 6**

Napisz algorytm który dla liczby naturalnej  $r$  sprawdza, czy rok  $r$  jest przestępny (zakładamy, że zawsze obowiązywał kalendarz gregoriański).