

# Algorytmy i struktury danych - procedury, funkcje, złożoność

Marcin Żurowski

31 marca 2022

# Plan zajęć

- 1 Pseudokod
- 2 Złożoność
- 3 Notacje asymptotyczne
- 4 Ćwiczenia
- 5 Zadania domowe

# Procedura

```
procedure N(PF)
```

```
  S
```

```
  I
```

Przykład:

```
procedure MIN(a,b)
```

```
  integer a, b
```

```
  if a < b
```

```
    MIN = a
```

```
  else
```

```
    MIN = b
```

# Procedura

```
procedure N(PF)
```

```
  S
```

```
  I
```

Przykład:

```
procedure MIN(a,b)
```

```
  integer a, b
```

```
  if a < b
```

```
    MIN = a
```

```
  else
```

```
    MIN = b
```

# Wejście i wyjście

- `read(a,b)`
- `write(c)`

# Wejście i wyjście

- `read(a,b)`
- `write(c)`

# Potęga iteracyjnie

```
POTEGA(a, n)
```

```
  a ∈ ℝ
```

```
  n ∈ ℕ
```

```
  pot = 1
```

```
  for i = 1 to n
```

```
    pot = pot * a
```

```
  return pot
```

# Potęga binarnie

```
POTEGA(a, n)
```

```
  a ∈ ℝ
```

```
  n ∈ ℕ
```

```
  i = n
```

```
  pot = 1
```

```
  pod = a
```

```
  while i ≠ 0
```

```
    if i MOD 2 ≠ 0
```

```
      pot = pot * pod
```

```
    pod = pod * pod
```

```
    i = i DIV 2
```

```
  return pot
```



# Szacowanie funkcji

$$\exists n_0 \in \mathbb{N} \forall n \geq n_0 1 \leq \lg n \leq n \leq n \lg n \leq n^2 \leq n^3 \leq \dots \leq n^k \leq \\ \leq \dots \leq 2^n \leq n! \leq n^n$$

# Notacje asymptotyczne

- $O(g(n)) = \{f(n) : \exists c > 0 \exists n_0 \in \mathbb{N} \forall n \geq n_0 0 \leq f(n) \leq cg(n)\}$
- $\Theta(g(n)) = \{f(n) : \exists c_1 > 0 \exists c_2 > 0 \exists n_0 \in \mathbb{N} \forall n \geq n_0 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)\}$
- $\Omega(g(n)) = \{f(n) : \exists c > 0 \exists n_0 \in \mathbb{N} \forall n \geq n_0 0 \leq cg(n) \leq f(n)\}$

# Zadania

1  $f(n) = 13n^2 + 4n - 7$

2  $f(n) = n^2 + 1$

3  $f(n) = 2n^4 + 3n - 8$

4  $f(n) = n^3 + 3n - 1$

5  $f(n) = n^3 + 5$

6  $f(n) = n^2 + n + 1$

7  $f(n) = 3^n$

8  $f(n) = 2^{n+1}$

9  $f(n) = 2^{2n}$

10  $f(n) = (n + 1)^2$

11  $f(n) = (200n)^2$

Podaj definicję procedury wyznaczającej najmniejszą wartość tablicy  
A

W tablicy  $A$  zawierającej liczby znajdują się dwa uporządkowane (niemalejąco) ciągi na miejscach od indeksu  $p$  do indeksu  $q$  i od indeksu  $q + 1$  do indeksu  $r$ , gdzie  $p \leq q < r$ . Zapisać definicję procedury `scal`, która scali te dwa ciągi w jeden niemalejący ciąg i umieści go w tablicy  $A$  na miejscach od  $p$  do  $r$ .

Zilustruj działanie procedury `scal` na konkretnym przykładzie.  
Dobierz tak elementy tablicy `A`, aby pokazać działanie wszystkich pętli `while`

Zadana jest tablica  $A[p..r]$  zawierająca liczby. Napisać definicję procedury `podziel`, która podzieli tę tablicę (poprzez przestawienie jej elementów) na dwie tablice,  $A[p..q - 1]$  oraz  $A[q + 1..r]$ , w ten sposób, że każdy element z pierwszej podtablicy jest nie większy niż element  $A[q]$ , który z kolei jest mniejszy od każdego elementu z drugiej podtablicy. Obliczenie indeksu  $q$  ma stanowić część tej procedury podziału.

Zilustruj działanie procedury `podziel` dla tablicy `A[3..10]` zawierającej dowolnie wybrane liczby całkowite.



Jaką wartość  $q$  oblicz procedura `podziel`, gdy wszystkie elementy  $A[p..r]$  mają taką samą wartość?

Jaką wartość  $q$  oblicz procedura `podziel`, gdy elementem rozdzielającym jest najmniejszy element tablicy  $A[p..r]$ ?

Dane są dwa wektory  $A[1..n]$  i  $B[1..n]$  zawierające liczby. Napisz definicję funkcji logicznej, przyjmującej wartość true wtedy i tylko wtedy, gdy oba wektory są równe, to jest

$$A[i] = B[i] \text{ dla } i = 1, \dots, n$$

Zapisać definicję funkcji wyznaczającej w tablicy dwuwymiarowej liczbę największych elementów