

Algotrymy i struktury danych

Lista 2

Zadanie 1.

Wylicz ile linii wypisze poniższy program (podaj wynik będący funkcją od n w postaci asymptotycznej $\Theta(\cdot)$). Można założyć, że n jest potęgą 3.

```
1: function f(n)
2:   if n > 1 then
3:     print_line('still going')
4:     f(n/3)
5:     f(n/3)
6:   end if
```

Zadanie 2.

Niech $f(n)$ i $g(n)$ będą funkcjami asymptotycznie nieujemnymi (tzn. nieujemnymi dla dostatecznie dużego n). Korzystając definicji notacji Θ udowodnij, że $\max\{f(n), g(n)\} = \Theta(f(n) + g(n))$.

Zadanie 3.

Wylicz asymptotyczną złożoność (używając notacji Θ) poniższych fragmentów programów:

```
1: for i = 1 to n do
2:   j = i
3:   while j < n do
4:     sum = P(i, j)
5:     j ++
6:   end while
7: end for
```

```
1: for i = 1 to n do
2:   j = i
3:   while j < n do
4:     sum = R(i, j)
5:     j = j + j
6:   end while
7: end for
```

gdzie:

- koszt wykonania procedury $P(i, j)$ wynosi $\Theta(1)$
- koszt wykonania procedury $R(i, j)$ wynosi $\Theta(j)$.

Zadanie 4.

Wyznacz asymptotyczne oszacowanie górne dla następujących rekurencji:

- $T(n) = 2T(n/2) + 1$
- $T(n) = 2T(n/2) + n$
- $T(n) = 3T(n/2) + n \log(n)$

Zadanie 5.

Zaprojektuj algorytm wczytujący z wejścia tablicę liczb $A[1], \dots, A[N]$ i przygotowujący tablicę B taką, że na jej podstawie będzie potrafił odpowiadać na pytania: ile wynosi suma elementów tablicy A od miejsca i do miejsca j włącznie dla $i < j$. Jaka jest złożoność czasowa Twojego algorytmu? Ile pamięci zajmuje tablica B ? Ile zajmuje odpowiedź na jedno pytanie?

Zadanie 6.

Pokaż, jak grać w grę w "10 pytań", w której wiadomo, że wybrana liczba jest dodatnia, ale nie jest na początku znane górne ograniczenie jej wartości. Ile pytań potrzebujesz, żeby zgadnąć dowolną liczbę (liczba pytań może zależeć od wielkości liczby)?

Zadanie 7.

Używając algorytmu "divide-and-conquer" do mnożenia liczb wykonaj mnożenie dwóch liczb binarnych 11011, 1010.