

Wykład10, złożoność obliczeniowa algorytmów

Rafał Grot

January 13, 2023

Contents

1	DLL TSP	1
2	Złożoność obliczeniowa algorytmów	2
2.1	Złożoność pamięciowa	2
2.2	Złożoność czasowa	2
2.3	Typowe funkcje złożoności obliczeniowej	2
2.3.1	Funkcja Stała	2
2.3.2	Funkcja liniowa	2
2.3.3	Funkcja kwadratowa	3
2.3.4	Funkcja wielomianowa	3
2.3.5	Funkcja wykładnicza	3
2.3.6	Funkcja silnia wykładnicza	3
3	P, NP	3

1 DLL TSP

Nazwisko_Imię_GRXXX.dll

Tabela eksportu FindRoad.

A nie:

- _FindRoad
- _FindRoad@12
- FindRoad@12

Przed deklaracją funkcji exportowej `extern "C" void __stdcall FindRoad...`

- w zależności od kompilatora `__stdcall` trzeba wywalić.

`tdumb -ee nazwa.dll`

2 Złożoność obliczeniowa algorytmów

2.1 Złożoność pamięciowa

2.2 Złożoność czasowa

Jak szybko rośnie zapotrzebowanie algorytmu wraz ze wzrostem rozmiaru zadania.

N – rozmiar zadania algorytmicznego.

$f(N)$ – funkcja złożoności obliczeniowej.

2.3 Typowe funkcje złożoności obliczeniowej

2.3.1 Funkcja Stała

$$F(N) = A, A = \text{const}$$

$$O(1)$$

1. $O(1)$

$$O(1) = O(1) + O(1) + \dots + O(1)$$

$$O(1) = A + O(1)$$

2.3.2 Funkcja liniowa

$$F(N) = A \cdot N + B, A, B = \text{const}$$

$$\alpha = \text{tg } A$$

1. $O(N)$

$$O(N) = A \cdot O(N)$$

$$O(N) = O(N) + O(N) + \dots + O(N)$$

$$O(N) = O(1) \cdot O(N)$$

$$O(N) = N \cdot O(1)$$

2.3.3 Funkcja kwadratowa

$$F(N) = A \cdot N^2 + B \cdot N + C$$

1. $O(N^2)$

$$O(N^2) = A \cdot O(N^2)$$

$$O(N^2) = O(N^2) + O(N^2) + \dots + O(N^2)$$

$$O(N^2) = O(N^2) + O(N)$$

$$O(N^2) = O(N^2) + O(1)$$

$$O(N^2) = N \cdot O(N)$$

$$O(N^2) = O(1) \cdot O(N^2)$$

$$O(N^2) = O(N) \cdot O(N)$$

2.3.4 Funkcja wielomianowa

$$F(N) = A \cdot N^B + \dots + X, A, B, \dots, X = \text{const}$$

2.3.5 Funkcja wykładnicza

$$F(N) = A^N + B^{B_1} + \dots + X$$

2.3.6 Funkcja silnia wykładnicza

$$f(N) = N!$$

3 P, NP

Zadania klasy P , są to zadania które są rozwiązywalne przez algorytm w czasie wielominaowym, przez deterministyczną maszynę Turinga.