Kraków 21.10.2019

Zestaw zadań nr. 2

- Zadanie 1
  Zakładając że $f_1(n)$ jest $O(g_1(n))$ i $f_2(n)$ jest $O(g_2(n))$ udowodnij (wprost z definicji) następujące własnoci:
  - a) $f_1(n) + f_2(n) = O(max(g_1(n)), g_2(n)))$
  - b) Jeśli istnieje liczba k taka, że dla każdego $n > k$, $g_1(n) < g_2(n)$ to $f_1(n) + f_2(n) = O(g_2(n))$
  - c) $f_1(n) \cdot f_2(n) = O(g_1(n) \cdot g_2(n))$
  - d) $O(c \cdot g(n)) = O(g(n))$
  - e) c jest O(1)
- Zadanie 2
  Dla przykładowych funkcji zależności algorytmu od liczby danych n: $f(n) = 1000n^{50} + 2n^2$ i $g(n) = 0.0000001n^{50} + 665n$ oszacuj ich zlożoność oblicznioową. Czy spełniają warunek $f(n) = O(g(n))$?
- Zadanie 3
  Jaka jest złożoność obliczeniowa algorytmu sortowania przez wstawianie. Narysuj schemat blokowy i przeanalizuj ilość wykonywanych operacji.
- Zadanie 4
  Przeprowadź analizę czasu działania bloków programu:
  - Pętla *while*, *do while*, *for* (nie zawierających wywołań funkcji)
  - Instrukcja *for* sekwencyjnego bloku instrukcji
  - Czas działania programu zawierającego wywołanie funkcji
  - Czas działania bloku zawierającego funkcje rekurencyjne
- Zadanie 5
  Przeanalizuj ilość wykonywanych operacji dla algorytmu wyszukującego liczbę w tablicy. Narysuj schemat blokowy. Które operacje są dominujące? Jak jest średnia, optymistyczna i pesymistyczna zlożoność obliczeniowa tego algorytmu.
- Zadanie 5
  Rozważmy problem wykrywania powtarzających się elementów w ciągu n liczb $< x_1, x_2, ..., x_n >$. Pokaż, że można rozwiązać ten problem w czasie $\Theta(n\ lg_2 n)$.
  Wskazówka: Przyjmij że sortowanie możesz wykonać w czasie $\Theta(n\ lg_2 n)$.
- Zadanie 6
  Uporządkuj podane niżej funkcje wg. asymptotycznego stopnia złożoności tak, aby każda funkcja była asymptotycznie mniejsza od następujących po niej:
  51n+101, $\frac{n^3}{7lg^7 n}$, $\frac{n^2+2}{lgn}$, $(\sqrt{n}+1)^3$, $\frac{lgn}{n}$, $\frac{n}{lgn}$, $\sum_{k=0}^{n} k\sqrt{k}$.

- Zadanie 7

  Korzystając z twierdzenia o rekursji uniwersalnej oszacuj rząd wielkości funkcji T zadanej równaniem rekurencyjnym:

  - $T(n) = 4T(\frac{n}{2}) + n$
  - $T(n) = 4T(\frac{n}{2}) + n^2$
  - $T(n) = 4T(\frac{n}{2}) + n^2 lg_2 n$
  - $T(n) = 4T(\frac{n}{2}) + n^3$
  - $T(n) = T(\frac{n}{2}) + c$