1 Lista 3, Zadanie 9

Zadanie polega na przeanalizowaniu i określeniu całkowitej złożoności algorytmów rekurencyjnych, dla których określony jest podział na ilość pod-problemów, rozmiar pod-problemu i koszt scalania.

1.1 Algorytm A

$T(n) = 5T(\frac{n}{2}) + O(nlog(n))$

Jako, że mamy bardzo elegancko zadane współczynniki możemy skorzystać *Twierdzenia o Rekurencji Uniwersalnej*. U nas:

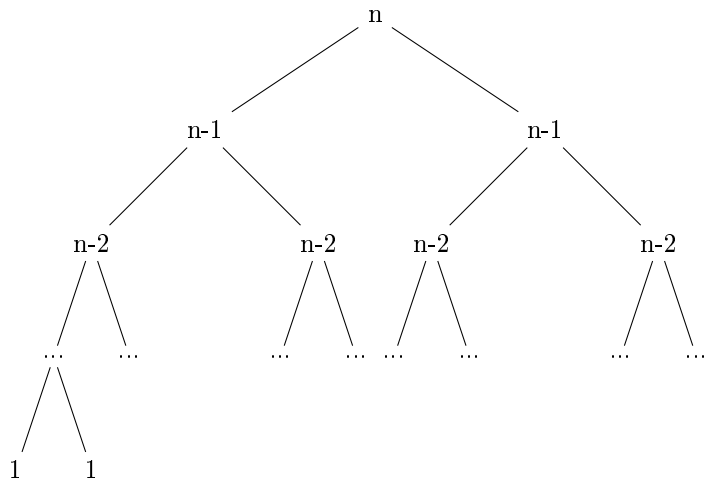
$$a = 5$$
$$b = 2$$
$$\log_2 5 \approx 2.32$$

d z całą pewnością jest mniejsze od 2.32 zatem $T(n) = O(n^{\log_2 5})$

1.2 Algorytm B

$T(n) = 2T(n-1) + O(n)$

W tym przypadku nie jest już tak fajnie. Algortym nie dzieli problemu na problemy k-razy mniejsze a na o 1 mniejsze. Przez to nie możemy skorzystać z *Twierdzenia o Rekurencji Uniwersalnej*.
Przeanalizujmy koszt algorytmu na kolejnych poziomach drzewa binarnego. (Wartości węzłów są kosztami).



Drzewo jest pełne i kończy się na liściu o wartości 1. Nietrudno zauważyć, że wysokość drzewa to n.

A więc na k-tym poziomie drzewa koszt poziomu wynosi $2^k(n-k)$ W takim razie $T(n) = \sum_{k=0}^{n} 2^k(n-k) = 2^{n+1} - n - 2 = O(2^n)$. Niedobrze.

1.3 Algorytm C

$T(n) = 9T(\frac{n}{3}) + O(nlog(n^2))$

Ponownie możemy zastosować *Twierdzenie o Rekurencji Uniwersalnej*. U nas:

$$a = 9$$
$$b = 3$$
$$\log_3 9 = 2$$
$$d = 2$$

Bardzo porządny algorytm. $d = \log_3 9$, bo $2 = 2$ zatem $T(n) = O(n^2 \log(n))$

1.4 Wybór

Algorytm B nie wchodzi w grę, jego złożoność obliczeniowa rośnie szybciej niż ceny alkoholu przed wprowadzeniem prohibicji.
Porównując algorytm A i C, ze złożonościami $O(n^{\log_2 5}) \approx O(n^{2.32})$ i $O(n^2 \log(n))$ możemy skorzystać z tego, że dowolny wielomian rośnie szybciej niż logarytm innymi słowy $n^{2.32} > n^2 log(n)$ już dla stosunkowo niedużych n-ów. Wybrałbym więc algorytm C.