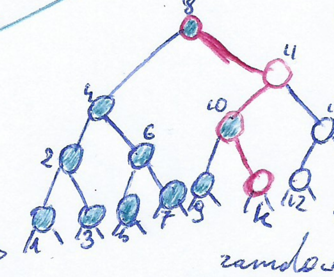
Lista 6, zadanie 8

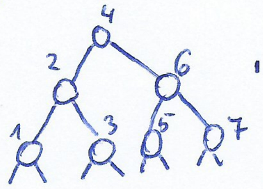
Do operacji split będziemy potrzebować operacji join. join(t_1, k, t_2) działa w czasie $O(|h(t_1)-h(t_2)|)$ i zwraca drzewo t takie, że $\max(h(t_1), h(t_2)) \leq h(t) \leq \max(h(t_1), h(t_2)) + 1$. Dowód jest oczywisty więc go pominę

Aby wykonać split(T, k), musimy znaleźć miejsca podziału. Możemy to zrobić np. wstawiając k do T. Wtedy wszystkie elementy na lewo od ścieżki, po której szliśmy wstawiając są mniejsze od k a na prawo mamy pozostałe elementy. Koszt tego to $O(\log n)$.

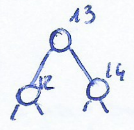


przykładowe drzewo, na czerwono mamy ścieżkę po której szliśmy, elementy zamalowane znajdują się na lewo od k

Teraz dostajemy dużo poddrzew, które musimy połączyć w dwa drzewa - T_L i T_R. W przypadku z rysunku będzie tak:

T_{L1} = join( , 8, join(9, 10, ⊖))

pusty liść, syn k

T_{R1} = join(⊖, 11, ).

Rozważmy tylko złożoność dla T_L, dla T_R będzie analogicznie. Załóżmy, że po podziale mamy l drzew - $T_1, T_2, \ldots, T_l$ takich, że $h(T_1) \leq h(T_2) \leq \ldots \leq h(T_l)$. Teraz łączymy po kolei: T_1 z T_2, i z tego otrzymamy T_2', T_2' łączymy z T_3 i otrzymamy T_3' itd aż połączymy wszystkie drzewa. Złożoność tego to koszt $l-1$ operacji join. Z operacji join wiemy, że

$h(T_i') \geq h(T_i)+1$, czyli $h(T_i') \leq h(T_i)+1 \leq h(T_{i+1})+1$. Koszt jednej operacji to $O(|h(T_{i+1})-h(T_i')|)$. Suma wszystkich kosztów to $\sum_{i=2}^{l-1} |h(T_{i+1})-h(T_i')| + |h(T_2)-h(T_1)|$ Dla uproszczenia zapisu niech $T_1' = T_1$, przy czym T_1' spełnia zależność wyżej. Wtedy mamy

$$\sum_{i=1}^{l-1} |h(T_{i+1})-h(T_i')| \leq \sum_{i=1}^{l-1} h(T_{i+1})-h(T_i')+2 = (l-1)\cdot 2 + \sum_{i=2}^{l-1}(h(T_i)-h(T_i')) + h(T_l)-h(T_1') = 2l-2+h(T_l)-h(T_1)+\sum_{i=2}^{l-1}(h(T_i)-h(T_i'))$$

, a ponieważ wiemy, że $\max(h(T_i), h(T_{i-1}')) \leq h(T_i') \leq \max(h(T_i), h(T_{i-1}'))+1$, to $2l-2+h(T_l)-h(T_1)+\sum_{i=2}^{l-1}(h(T_i)-h(T_i')) \leq 2l-2+h(T_l)-h(T_1)+l-1 = 3l-3+h(T_l)-h(T_1)$. Teraz zauważmy, że $l \leq h(T)$. Wynika to z tego, że „przecinając" drzewo wstawieniem do niego k, w każdym zejściu poziom niżej tworzymy tylko jedno nowe poddrzewo do połączenia. Zauważmy też, że w ostatnim kroku tworzymy dwa drzewa, z których większe jest poddrzewem oryginalnego T. Czyli $h(T_l) \leq h(T)-1$. Stąd mamy złożoność $3l-3+h(T_l)-h(T_1) \leq 3h(T)+h(T)-4-h(T_1) = O(h(T))$, czyli $O(\log n)$ □