l4z1

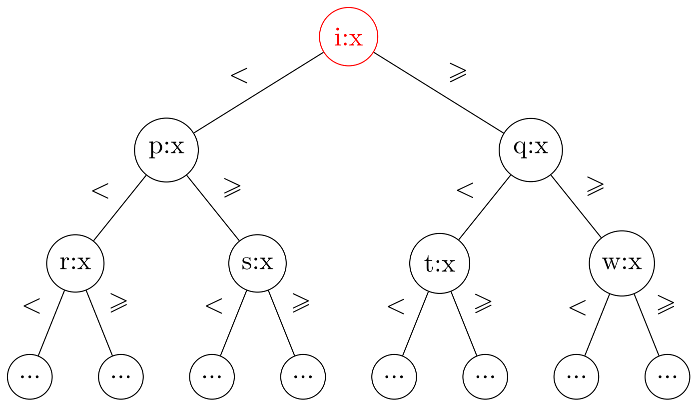
21 kwietnia 2020

1 Treść zadania

Rozważ wyszukiwanie elementu x w posortowanej tablicy $A[1, ..., n]$ o różnych elementach. Wiemy, że może zostać do tego użyty algorytm Binary-Search, który ma złożoność obliczeniową $O(\log n)$. Pokaż, że w "comparison model" (czyli można zadawać tylko pytania w stylu: czy $A[i] \geqslant z$?), wyszukiwane jest $\Omega(\log n)$.

2 Rozwiązanie

Mamy tablicę $A[1, ..., n]$ i wyszukujemy w niej element x. Przyjmijmy, że mamy algorytm, który zwraca i jeśli $x \leqslant A_i$, oraz 0 w przeciwnym wypadku. Rozważmy następujące drzewo decyzyjne:



(1)

Drzewo decyzyjne musi mieć tyle liści, ile różnych wartości może być zwróconych (x może być na dowolnej pozycji w tablicy)

Nasze drzewo decyzyjne musi mieć jedeń liść który jest wynikiem porównania x z A_i dla $i = 1, ..., n$, czyli musi mieć n liści. W przeciwnym wypadku można skonstruować taką tablicę A, dla której algorytm podawałby niepoprawny wynik. Drzewo o wysokości k może mieć co najwyżej 2^k liści. Otrzymujemy zatem następującą nierówność:

$$2^k \geqslant n$$
$$\log 2^k \geqslant \log n$$
$$k \geqslant logn$$

Drzewo o wysokości k reprezentuje k różnych porównań elementu x z pewnymi elementami A. Pokazaliśmy zatem, że ilość porównań musi być większa lub równa $\log n$, co bezpośrednio daje nam, że wyszukiwanie jest $\Omega(\log n)$.

W ogólności myślimy o dowolnym algorytmie, więc teoretycznie on nie zawsze musi porównywać x z jakimś elementem A. Niemniej jednak w każdym kroku wykonuje porównanie pewnych dwóch wartości.