L5 Z2

Jak wyznaczyć i-ty następnik zadanego węzła x w drzewie statystyk pozycyjnych w czasie $O(\log n)$, gdzie n oznacza rozmiar drzewa.

Najpierw znajdujemy rangę węzła x. Zaczynając od x i idąc w górę po kolejnych rodzicach, zliczamy elementy o randze niższej od x. Jeśli dany węzeł jest prawym potomkiem, to wszystkie elementy w poddrzewie zaczynającym się od jego lewego brata mają rangę niższą niż on.

Algorithm 1: Znajdź rangę x

```
r = x.left.size + 1
while x ≠ root do
    if x = x.p.right then
        r = r + x.p.left.size + 1
    end
    x = x.p
end
r to ranga naszego węzła
```

To oznacza, że wartość w węźle x jest r-tą statystyką pozycyjną zbioru przechowywanego w RB-drzewie

Ilość iteracji w pętli WHILE zależy od wysokości drzewa. wysokość drzewa czerwono-czarnego to max 2log(n), złożoność tego algorytmu to O(log n).

nieco bardziej precyzyjnie: 2 log(n+1)

Następnie szukamy i-ty następnik węzła x, czyli węzeł o randze i większej od x. Wyznaczyliśmy rangę węzła x, zatem szukamy w drzewie węzła o randze $r + i$.

Algorithm 2: Znajdź element o zadanej randze i w poddrzewie zaczepionym w węźle x

```
Function Select(x, i):
    r = x.left.size + 1
    if i == r then
        return x
    else if i<r then
        return Select(x.left, i)
    else
        return Select(x.left, i-r)
```

Wywołujemy Select($root$, $r + i$) i otrzymujemy szukany następnik węzła x.

Złożoność algorytmu Select również jest zależna logarytmicznie od wysokości drzewa. Maksymalny czas działania obu algorytmów razem to 4log(n), Minimalny to 2log(n), bo w drzewie czerwono-czarnym najdłuższa możliwa ścieżka ma długość 2log(n), a najkrótsza log(n). Możemy znaleźć szukany następnik w O(log n)

Znalezienie i-tego następnika w drzewie statystyk pozycyjnych może być wykonane trochę bardziej efektywnie niż w przedstawionym rozwiązaniu, gdzie w procedurze znajdowania rangi elementu (OS-Rank) i procedurze znajdowania elementu o zadanej statystyce pozycyjnej (OS-Select) algorytm przechodzi zawsze od startowego węzła 'x' do korzenia, a następnie od korzenia do szukanego węzła 'y'.

Natomiast to często nie jest konieczne, np. jeśli i <= x.right.size to i-ty następnik 'x'-a znajduje się w jego prawym poddrzewie i wystarczy go tam znaleźć przy użyciu OS-Select. Może się zdarzyć sytuacja, ze musimy wejść kilka poziomów ponad 'x' (odpowiednio przy tym modyfikując szukaną statystykę pozycyjną), a następnie odpalić OS-Select na poddrzewie dla którego zachodzi i_new <= y.size.

Oczywiście w worst case to jest nadal algorytm, który będzie miał złożoność podobną do tej w podanym powyżej rozwiązaniu, czyli O(log(n)).