Zadanie 8

Treść

Napisz procedurę *split(T, k)* rozdzielającą drzewo AVL T na dwa drzewa względem klucza k.

Rozwiązanie

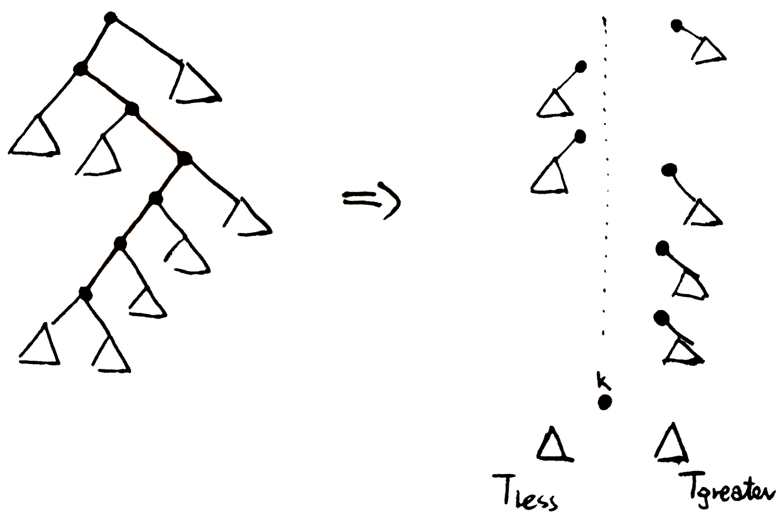
Idea: Podzielimy drzewo na mniejsze drzewa. Niektóre zawierające klucze większe niż k, niektóre zawierające klucze mniejsze lub równe k. Następnie połączymy powstałe drzewa, tak aby otrzymać dwa drzewa, które będą wynikiem.

```
split(T, k) :
  // listy, na których będziemy pamiętać pary (drzewo, wierzchołek)
  glist -- drzewa zawierające klucze większe niż k
  llist -- drzewa zawierające klucze mniejsze równe niż k

  // rozbijanie drzewa na wiele drzew

  v = T.root
  while v != leaf || v.key != k :
    if v.key > k :
      glist.add(v.right, v)
      v = v.left

    if v.key < k :
      llist.add(v.left, v)
      v = v.right

  less = v.left
  greater = v.rigth
  // jesli v == leaf to left == right == puste drzewo

  // łączenie drzew w drzewa wynikowe

  while !glist.empty() :
    t, v = glist.front()
    glist.pop()
    greater = join(greater, t, v)

  while !llist.empty() :
    t, v = llist.front()
    llist.pop()
    less = join(less, t, v)

    return left, right
```

Procedura `join(T1, T2, v)` to standardowa procedura łącząca dwa drzewa AVL używając do tego wierzchołka v. Jej złożoność to $O(h)$ gdzie h - różnica wysokości drzew.

Analiza złożoności

Podczas rozdzielania drzewa dostaniemy w sumie $O(\log n)$ drzew, po jednym na każdym poziomie (patrząc od korzenia wyjściowego drzewa) oraz na samym końcu dwa drzewa na tym samym poziomie, które oznaczę T_{less} i $T_{greater}$. Dodatkowo jeśli w drzewie znajduje się klucz względem którego dzielimy drzewo, to zostanie również jeden wierzchołek z tym kluczem. Koszt rozdzielania drzewa to $O(\log n)$.



Zastanówmy się teraz ile operacji wykonujemy łącząc dwa drzewa, których wysokości różnią się o k.

Przy łączeniu drzew wykorzystujemy standardową procedurę, która przyjmuje dwa drzewa oraz wierzchołek, który pomaga w złączeniu tych drzew. BSO założę, że łączymy drzewa T_1 i T_2 o wysokościach $h_1 > h_2$.

```
1. Schodzimy zwasze do prawego poddrzewa, dopóki wielkość poddrzewa jest większa o
   więcej niż 2 od drzewa T2
2. Łączymy drzewo T2 z poddrzewem, do którego doszliśmy i podczepiamy pod jego
   rodzica.
3. Mógł się zaburzyć porządek drzewa więc musimy wykonać naprawianie drzewa w górę
```

W takim razie złączenie drzew o różnicy wysokości k wykonujemy w czasie $O(k)$.

Zauważmy, że łącząc otrzymane listy po kolei w drzewa wynikowe, wykonamy $O(\log n)$ operacji łączenia (złożoność każdej z nich, możemy oszacować przez $O(\log n)$, ponieważ różnicę wysokości możemy oszacować przez wysokość całego drzewa). W ten sposób otrzymujemy oszacowanie kosztu $O(\log^2 n)$.

Można pokazać, że ta złożoność to faktycznie $O(\log n)$. Dostajemy to dzięki temu, że mamy po jednym drzewie na każdym poziomie.

Rozważmy ciąg operacji łączenia, które będziemy wykonywać po tej samej stronie (BSO będziemy łączyć l razy z drzewem T_{less}.

Dwa sąsiednie drzewa z listy (które różnią się jednym poziomem licząc od korzenia drzewa) mogą różnić się wysokością o 0, 1, 2 lub 3 oraz wysokość drzewa wrzuconego na listę wcześniej będzie większa równa wysokości drzewa wrzuconego później. (W ogólnym przypadku kolejne drzewa z tej samej listy różniące się poziomem o k mogą różnić się maksymalnie o k, $k + 1$, $k + 2$ lub $k + 3$.)

Przeanalizujmy teraz ile operacji w sumie wykonamy podczas łączenia listy drzew.

Przyjmijmy, że każda operacja łączenia wpłaca $2 \cdot 2 \cdot 3$ kredytów. Kredyty będziemy przypisywać po 2 trzem kolejnym poziomom drzewa licząc od poziomu drzewa ściąganego w danym kroku z listy. Podczas łączenia drzew wykonujemy 2 operacje na każdym z kolejnych poziomów (ich liczba zależy od różnicy wysokości). Teraz zauważmy, że operacja łącznia drzew różniących się o nie więcej niż 3 wykorzysta kredyty tylko z poziomów, na które wpłaci kredyty, zostawiając tym po 2 kredyty na każdym z tych poziomów.

W ten sposób ciąg operacji na sąsiednich elementach z listy po jednej stronie zostawi przynajmniej po 2 kredyty na poziomach pośrednich. W ten sposób operacja łączenia drzew po drugiej stronie (która będzie miała do połączenia drzewa różniące się poziomami o więcej niż 1) będzie miała wystarczająco dużo kredytów na poziomach pośrednich.

W ten sposób otrzymujemy zamortyzowany czas stały operacji złączania drzew.

W takim razie cała operacja rozdzielania drzewa będzie miała złożoność $O(\log n)$.