Treads (Drzewiec)

piątek, 17 maja 2024 11:13

Dane: klucze, priorytety

$(k_1, p_1)\ (k_2, p_2) \dots (k_n, p_n)$

- BST wg wartości $k_1, \dots, k_n$
- kopiec wg wartości $p_1, \dots, p_n$

Założenie
k_i są parami różne
p_i też są

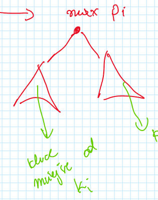
Przy tym założeniu drzewiec istnieje i jest unikalny

Operacje
- insert (k, p) – insert (k) jak w BST, rotacjami przywracamy porządek kopcowy
- delete (k) – rotacjami przesuwamy k na dół, gdy stanie się liściem usuwamy

rotujemy syna o większym priorytecie

Dla KAŻDEGO zestawu kluczy i priorytetów możemy utworzyć drzewiec!

→ w korzeniu max p_i → max p_i



klucze mniejsze od k_i

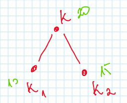
klucze większe od k_i

Gdy klucze są unikalne, to drzewiec jest jednoznacznie wyznaczony

insert (x)
losuj p dla x
wstaw x do drzewa T jak do BST → $O(\log n)$
przywróć porządek kopcowy
(bo rotacje to zachowują, więc się nie zmieni BST)

find (x) → jak w BST → $O(\log n)$ wysokość drzewa

delete (x) {
find(x)
rotuj x do liścia ← razem z tym który ma większe p
usuwam
↓
też coś $O(\log n)$
}

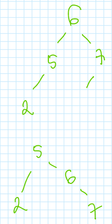


insert
1° jak w BST
2° losujemy dla k priorytet
3° rotacjami wrzucamy k na właściwe miejsce

każda rotacja zachowuje porządek BST

1, 2, 3
4 2 2

5, 6, 7, 2,
5 5, 3, 1



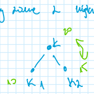
losowy drzewiec:
priorytety są losowane
(zał. ↙ z jednakowym ppb z $[0,1]$,

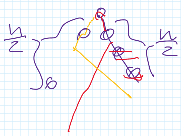
σ : $op_1, \dots, op_n$
- Przeanalizujemy oczekiwany koszt operacji delete
Niech kluczami będą $\{1, 2, \dots n\}$
Chcemy wykonać operację delete (m)

Założenie
Jeśli uporządkujemy klucze malejąco wg priorytetów to każda ich permutacja jest jednakowo prawdopodobna

Obserwacje:
Aby k przesunąć na pozycję liścia trzeba wykonać $r + l$ rotacji
gdzie r – długość skrajnie prawej ścieżki w lewym poddrzewie k
l – — " — lewej – " — w prawym — " —

rotujemy zawsze z większym synem względem priorytetu

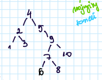




Oszacujemy oczekiwaną długość ścieżki (nazwijmy ją A) z korzenia do m.

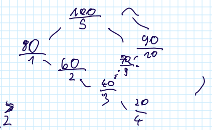
Przykład
Niech $n = 10$ → liczba kluczy
$m = 8$ → klucz do usunięcia

permutacja wg priorytetów: 4, 5, 9, 2, 1, 7, 3, 10, 8, 6
największy korzeń



Oznaczenie
$m_\leq = \{1, 2, \dots, m\}$
$m_\geq = \{m, m+1, \dots, n\}$

wartość oczekiwana długości ścieżki
$E(|A|) = E[|A \cap m_\leq|] + E[|A \cap m_\geq|] - 2$



Spostrzeżenie
Z kluczy $m_\leq$ na ścieżce A znajdą się te, które są maksimami prefiksowymi permutacji $\sigma_\leq$ obciętej do elementów z $m_\leq$

H_m – zmienna losowa równa liczbie maksimów prefiksowych w $\sigma_\leq$

$$x_i = \begin{cases} 1, & \text{jeśli 1 jest maksimum prefiksowym} \\ 0, & \text{wpp} \end{cases} \quad \text{dl } i = 1, \dots, m$$

$n = 10$
$m = 8$
$\sigma_\leq = \underline{3}\ 2\ \underline{4}\ 1\ \underline{7}\ 6\ \underline{8}\ 5$ – maks. pref.

$a T\left(\frac{n}{b}\right) + c$

$a \cdot f < 1$

$T(n) = T(\log n)$

$T\left(2^{2^k}\right) = 2^{2^{k-1}}$

$H_m = \sum_{i=1}^{m} x_i$

$$E[H_m] = E\left[\sum_{i=1}^{m} x_i\right] = \underset{\frac{1}{m}}{E[x_1]} + \overset{E[H_{m-1}]}{E\left[\sum_{i=2}^{m} x_i\right]} = (*) =$$

$Pr[x_1 = 1] = Pr$ [1 jest na pierwszym miejscu w $\sigma_\leq$ naszej permutacji] $= \frac{1}{m}$ bo tyle jest elementów

prawdopodobieństwo

Pozycja 1 nie ma wpływu na to czy i jest maksimum prefiksowym (dla $i = 2, \dots, m$)
Czyli liczbę $\sum_{i=2}^{m} x_i$ możemy liczyć maksima prefiksowe w permutacjach liczb $2, \dots, m$
a więc także w $1, \dots, m-1$ (bo możemy sobie zmniejszyć wszystkie indeksy o 1)

$E[H_m] = \sum_{i=1}^{m} \frac{1}{i} = O(\log m)$

Teraz chcemy obliczyć oczekiwaną liczbę rotacji potrzebnych do zsunięcia m do liścia

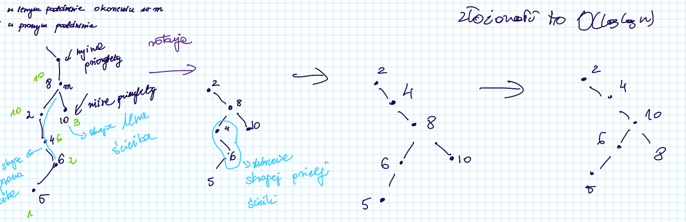
Obserwacja:
ta liczba jest równa długości skrajnie prawej ścieżki w lewym poddrzewie o korzeniu w m
+ długość skrajnie lewej ścieżki w prawym poddrzewie

zakładamy
Niech $n = 2^{2^k}$

$$T(n) = T\left(2^{2^k}\right) = T\left(\sqrt{2^{2^k}}\right) + O(1) = T\left(2^{2^k \cdot \frac{1}{2}}\right) + O(1) = T\left(2^{2^{k-1}}\right) + O(1) =$$

wykładnik zmalał

widzimy że wykona się k razy
skoro $n = 2^{2^k}$
to $k = \log\log n$
złożoność to $O(\log\log n)$



B – skrajnie prawa ścieżka w lewym poddrzewie

x_i – jak poprzednio

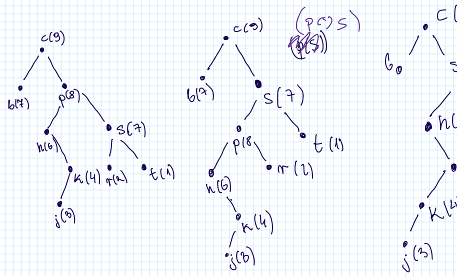
$$B = \sum_{i=1}^{m-1} x_i = E[x_1] + E\left[\sum_{i=2}^{m-2} x_i\right] = \frac{m-1}{m} < 1$$

$$E[x_1] = \frac{1}{m(m-1)} + \frac{1}{(m-1)(m-2)} + \dots + \frac{1}{2 \cdot 1}$$

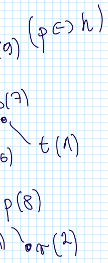
to dlatego, że to nie może być na pierwszym miejscu, musi być drugie

Analogicznie oczekiwana długość prawej ścieżki w poddrzewie o korzeniu w $m < 1$

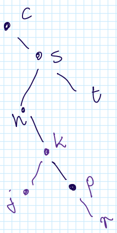
Stąd oczekiwana liczba rotacji < 2.



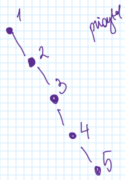
(9) $(p \leftrightarrow h)$



$p \leftrightarrow k$

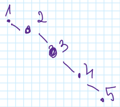


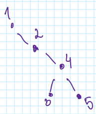
$p \leftrightarrow r$



klucze $1, \ldots, n$

priorytety $n, \ldots, 1$





$B = \sum_{i=1} x_i = E[X_1] + E[\sum_{i=2} x_i] = \frac{\dots}{m} < 1$ Analogicznie ...

$\frac{1}{m(m-1)} + \frac{1}{(m-1)(m-2)} + \dots + \frac{1}{2 \cdot 1}$

Stąd oczekiwana liczba rotacji < 2.

tu dochodzi o to że m nie leży na pierwszej ścieżce a 1 na drugiej

$m \quad 1 \quad \dots$

$\downarrow \quad \downarrow$

$\frac{1}{m} \cdot \frac{1}{m-1}$

Analogiczna struktura

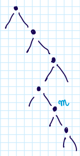
Oczekiwany koszt v2 delete:

Załóżmy, że nasze klucze to: $\{1, 2, \dots, n\}$

Chcemy usunąć klucz m

Delete (m)

1. odszukaj m $\quad O($długość ścieżki od korzenia do $m) \to O(\log m)$
2. zrotuj m do liścia $\quad$ skrajnie lewa i prawa ścieżka w prawym i lewym poddrzewie
3. usuń m $\quad O(1)$



Ozn.:

$m_{\leq} = \{1 \dots m\}$

$m_{\geq} = \{m, \dots, n\}$

A - ścieżka od korzenia do m

$|A|$ - długość ścieżki

Chcemy policzyć oczekiwaną długość ścieżki (po wszystkich losowaniach priorytetów)

$$E[|A|] = E[|A \cap m_{\leq}|] + E[|A \cap m_{\geq}|] - 2$$