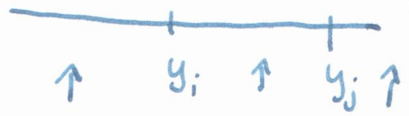


Prawdopodobieństwo tego, że y_i będzie porównane z y_j wynosi $\frac{2}{j-i+1}$.

Jeśli trafimy z pivotem na lewo od y_i lub na prawo y_j, to nic się nie dzieje, bo odkładamy decyzję o tym, czy będą porównane czy nie. Jeśli pivot jest wewnątrz przedziału y_i do y_j to elementy trafiają do różnych tablic i nie będą porównywane.

$$E[X_{ij}] = \frac{2}{j-i+1}$$

$$E[X] = \sum_{1 \leq i < j \leq n} \frac{2}{j-i+1} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j-i+1} \approx O(n \log n)$$

Modyfikacje:

- pivot - mechanika trzech losowych elem. i wybieramy środkowy z tych trzech
- jeśli w ciągu jest dużo powtórzeń to trój podział (mniejsze, równe i większe od pivota)
- stała pamięć

SELEKCJA

Problem:

Dane: $T[1 \ldots n]$ - ciąg, $k \in \mathbb{N}$ $(1 \leq k \leq n)$

Wynik: k-ty co do wielkości element

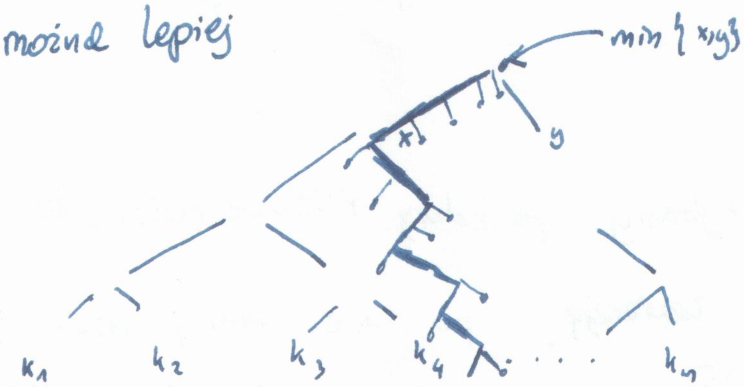
Łatwe przypadki:

1° $k=1$ koszt $\Theta(n)$

2° $k=2$

naiwnie: znajdź najmniejszy, usuń najmniejszy, znajdź najmniejszy

można lepiej



kandydaci na 2-go co do wielkości (jest ich $\lceil \log_2 n \rceil$)

spośród nich wybieremy elem. minimalny (#porówn $\lceil \lg n \rceil - 1$)

Pyt: czy można szybciej: Odp: NIE

Przypadek ogólny:

```
SELECTION(T, k):
    if |T| - małe then adhocmed(T,k)
    p ← jakiś elem z T
    U ← {x : x ∈ T & x < p}
    if (|U| ≥ k) then return SELECTION(U,k)
        else return SELECTION(T\U, k-|U|)
```

Algorytm magicznych piątek

T: c_1 c_2 c_3 … $c_{\lceil \frac{n}{5} \rceil}$

Pseudomed(T):

1° Podziel T na fragmenty 5-io elem

2° $\forall_{j=1\ldots\lceil \frac{n}{5} \rceil}$ $s_j \leftarrow$ mediana c_j

3° Niech $S = \{s_j\}_{j=1\ldots\lceil \frac{n}{5} \rceil}$

4° Return SELECTION$(S, \lceil \frac{|S|}{2} \rceil)$

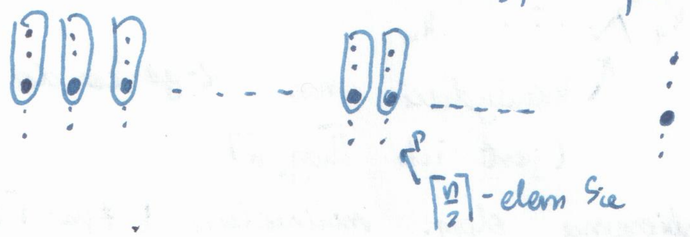
Tw.

Jeśli w procedurze SELECTION pivot jest wybierany procedurą PSEUDOMED to SELECTION wyznacza k-ty element ciągu T w czasie liniowym.

Poprawność algorytmu SELECTION jest oczywista

Lemat

Jeśli p jest wybrany procedurą PSEUDOMED, to zarówno U jak i $T \setminus U$ zawierają po nie mniej niż $\frac{3}{10}n-4$ elem.



W T elementów nie większych od p jest conajmniej

$3\left(\frac{1}{2}\lceil \frac{n}{5} \rceil - 1\right) \geq \frac{3}{10}n - 3$

Zatem elementów większych od p jest $\geq \frac{3}{10}n-4$

Koszt proc. SELECTION

$$K(n) \leq \begin{cases} \Theta(1) & \text{dla } n\text{-małego} \\ K(\lceil \frac{n}{5} \rceil) + K(\frac{3}{10}n+4) + \Theta(n) & \text{w.p.p.} \end{cases}$$

$K(n) = O(n)$ [to ma stałą koło 5]

Losowy wybór pivota

1° Algorytm Hoare'a

$p \leftarrow T[\text{rand}(1 \ldots n)]$

czas oczekiwany $O(n)$

2° Lazy select

1. Wybierz losowo, niezależnie, z powtórzeniami próbkę losową R złożoną z $n^{\frac{3}{4}}$ elem T
2. Posortuj R (w czasie $O(n^{\frac{3}{4}} \log n)$)
3. $L \leftarrow R[\frac{1}{2} n^{3/4} - \sqrt{n}]$

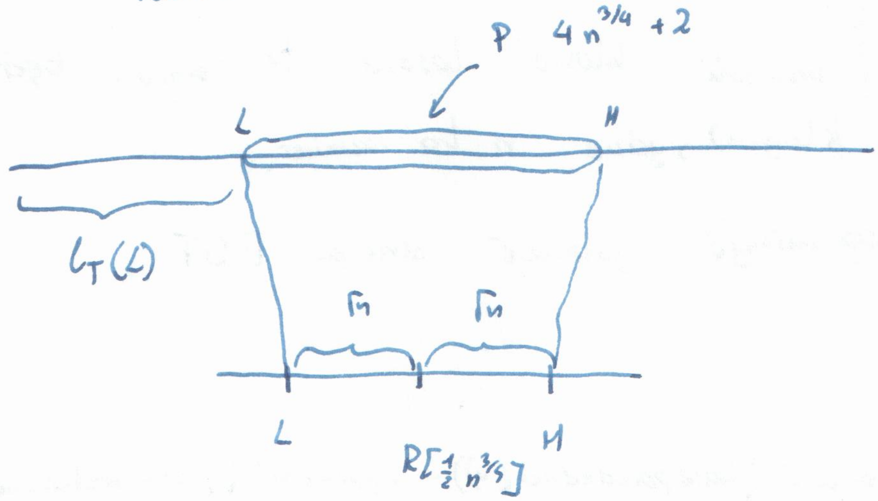
   $H \leftarrow R[\frac{1}{2} n^{3/4} + \sqrt{n}]$
4. Porównując L i H ze wszystkimi elementami T wyznacz:

   $$l_T(L) \leftarrow |\{y \in T : y < L\}|$$

   $$P \leftarrow \{y \in T : L \leqslant y \leqslant H\}$$
5. Sprawdź, czy
   - mediana zbioru T znajduje się w P, tj. $l_T(L) < \frac{n}{2} < l_T(L) + |P|$
   - $|P| \leqslant 4 \cdot n^{3/4} + 2$

   Jeśli nie, to PORAŻKA
6. Posortuj P

   return $P[\frac{n}{2} - l_T(L) + 1]$



FAKT:

Jeśli Lazy Select produkuje wynik to w czasie $O(n)$ i jest to wynik poprawny.

Aby oszacować prawdopodobieństwo porażki można skorzystać z Tw (nierówność Czebyszewa)!

Jeśli X jest zm. losową (o wartościach z $\mathbb{R}$) o wartości oczekiwanej μ_X i odchylenie standardowe σ_X, to $\forall_{t \in \mathbb{R}_+}$ $Pr[|X - \mu_X| \geq t\sigma_X] \leq \frac{1}{t^2}$

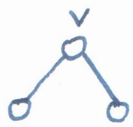
SŁOWNIKI

11.04.2018

operacje:
- insert
- delete
- find

BST

klucz(lewy_syn(v)) $\leq$ klucz(v) $\leq$ klucz(prawy_syn(v))



koszt operacji = O(wysokość drzewa)

Jak będziemy wrzucali klucze losowe to drzewo będzie miało wysokość $k\log(n)$, gdzie n to rozmiar.

CEL: chcemy ograniczyć wysokość drzewa BST

DRZEWA AVL

Niezmiennik $\forall_v$ |wysokość(lewe poddrzewo(v)) - wysokość(prawe poddrzewo(v))| ≤ 1

Fakt: Każde drzewo AVL o n wierzchołkach ma wysokość $O(\log n)$

Ozn. $g(h)$ = liczba pustych wskaźników w minimalnym drzewie AVL o wysokości h

tj. o najmniejszej # wierzchołków