CS3230 Chapter 6 Notes

Wednesday, 4 March 2020 12:35 PM

Order statistics: unsorted list of n elements, i^{th} order statistics is the i^{th} smallest element

- naive approach: merge sort, get ith smallest element.
- randomized divide and conquer:

rand-select(A[p..q], i) → ith smallest of A[p..q].
  if p=q return A[p].
  r = randpartition(A[p..q]).
  k = r-p+1.
  if i=k return A[r].
  if i<k return rand-select(A[p..r-1], i).
  else return rand-select (A[r+1..q], i-k).

- Best case: $T\left(\frac{9n}{10}\right) + \Theta(n)$, $n = \lg_{\frac{10}{9}} 1 = \Theta(n)$
- Worst case: $T(n-1) + \Theta(n)$, $= T(1) + \sum_{i=1}^{n} i = O(1) + \frac{n(n-1)}{2} = \Theta(n^2)$.

- Analysis of expected time:
  define the indicator random variable

$$X_k = \begin{cases} 1 & \text{if partition generates } k : n-k-1 \text{ split} \\ 0 & \text{otherwise} \end{cases}$$

To get upper bound, assume ith element always in larger side.
when $X_k = 1$, $T(n) = T(\max(k, n-k-1)) + \Theta(n)$.

$$\therefore T(n) = \begin{cases} T(\max(0, n-1)) + \Theta(n) & X_0 = 1 \\ T(\max(1, n-2)) + \Theta(n) & X_1 = 1 \\ \vdots & \\ T(\max(n-1, 0)) + \Theta(n) & X_{n-1} = 1 \end{cases}$$

$$= \sum_{k=0}^{n-1} T(\max(k, n-k-1)) + \Theta(n).$$

Calculating expectation,

$$E(T(n)) = E\left[\sum_{k=0}^{n-1} X_k \left(T(\max(k, n-k-1)) + \Theta(n)\right) \right]$$

$$= \sum_{k=0}^{n-1} E\left(X_k \cdot \left(T(\max(k, n-k-1)) + \Theta(n)\right)\right)$$

$k=0$ (by linearity of expectation).

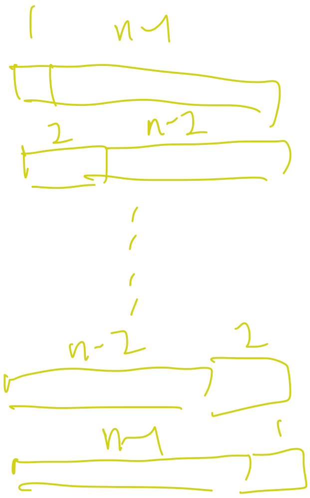
$$= \sum_{k=0}^{n-1} E[X_k] \cdot E[T(\max(k, n-k-1)) + \Theta(n)].$$

(by independence of X_k from other random choice).

$$= \frac{1}{n} \sum_{k=0}^{n-1} E[T(\max(k, n-k-1))] + \frac{1}{n} \sum_{k=0}^{n-1} \Theta(n).$$

(by linearity of expectation, $E[X_k] = \frac{1}{n}$

$$\leq \frac{2}{n} \sum_{k=\lfloor \frac{n}{2} \rfloor}^{n-1} E[T(k)] + \Theta(n)$$



$E[T(n)] \leq cn$ for constant $c > 0$.

$$\boxed{\sum_{k=\lfloor \frac{n}{2} \rfloor}^{n-1} k \leq \frac{3}{8} n^2.}$$

Substitution method:

$$E[T(n)] \leq \frac{2}{n} \sum_{k=\lfloor \frac{n}{2} \rfloor}^{n-1} ck + \Theta(n).$$

$$\leq \frac{2c}{n} \left(\frac{3}{8} n^2\right) + \Theta(n)$$

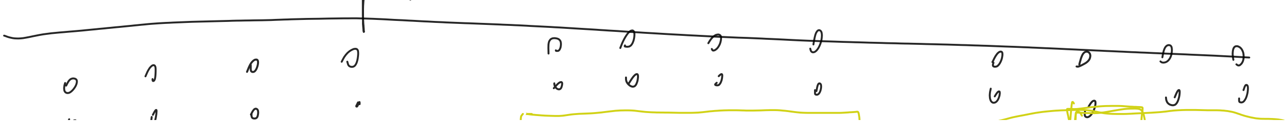
$$\leq cn.$$

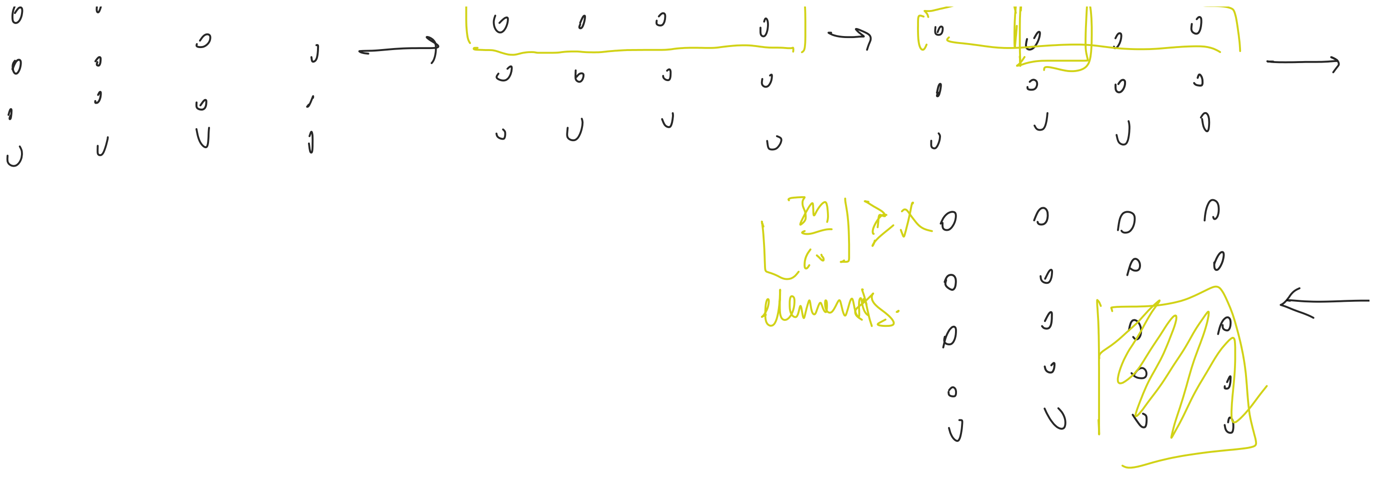
- works fast, but worst case is very bad. Generate a good pivot.
- select(i, n):
  - divide n elements into 5 groups. Find median of each group
  - recursively select median x of the $\lfloor \frac{n}{5} \rfloor$ group medians as pivot.
  - partition around pivot x.
  - if $i=k$ return x.

    elseif $i<k$.

    recursively select the ith smallest element in lower part.

    else recursively select $(i-k)$th smallest element in upper part.





when $n \geq 50$, $3\lfloor \frac{n}{10} \rfloor \geq \frac{n}{4}$.

For $n \geq 50$, select is executed recursively on

– Recurrence: $T(n) = T(\frac{1}{5}n) + T(\frac{3}{4}n) + c$

recursively select median x of the $\frac{n}{5}$ group medians to be pivot.

if recurs in else in

if divide into groups of 5:

- at least $\lfloor \frac{n}{10} \rfloor$ group medians $\geq x$
- at least $\lfloor \frac{n}{4} \rfloor$ elements $\leq x$ & at least
- recurse on $\lfloor \frac{3n}{4} \rfloor$ elements.

T

if divide into groups of 3:

- at least $\lfloor \frac{n}{6} \rfloor$ group median $\geq x$.
- at least $\lfloor 2\frac{n}{6} \rfloor$ group elements $\geq x$ & at l
  elements $\leq x$.
- recurse on $\lfloor n - \frac{n}{3} \rfloor$ elements $\rightarrow \frac{2n}{3}$.

- if divide into groups of 7:
  - at least $\lfloor \frac{n}{14} \rfloor$ group median $\geq x$.
  - at least $\lfloor \frac{4n}{14} \rfloor$ group elements $\leq x$ & a
    elements $\geq x$.
  - recurse on $\lfloor n - \frac{4n}{14} \rfloor$ elements $\longrightarrow \geq$