Divide & Conquer (Recursion)

Problem 1: (binary) Multiplication. of LARGE nos.

Given two n-bit numbers $A = a_{n-1}\, a_{n-2} \ldots a_0$
$B = b_{n-1}\, b_{n-2} \ldots b_0$

Compute $A \cdot B = C = c_{2n-1}\, c_{2n-2} \ldots c_0$

Ele. school Alg'm (... BC)

e.g.

```
      A= 1101
    * B= 1011
    ----------
         1101
        11010   <- shift of A by 1-bit
   +  ~~0000000~~
   + 1101000    <- Shift of A " 3-bits
   -------------
     10001111
```

Running Time $\rightarrow O(n)$

$\#_1(B)$ shifts

+ $\#_1(B)$ add.

each is $O(n)$ time

Total time: $O(n^2)$

CAN WE DO BETTER? YES!

Karastuba's Alg'm (1962):

Approach divide $A = \underbrace{a_{n-1} \ldots a_{n/2}}_{A'}\ \underbrace{a_{n/2-1} \ldots a_0}_{A''}$

$B = \underbrace{b_{n-1} \ldots b_{n/2}}_{B'}\ \underbrace{b_{n/2-1} \ldots b_0}_{B''}$

e.g. $A = 1101$ $n=4$

$A' = 11$, $A'' = 01$ $n/2 = 2$

$B = 1011$

$B' = 10$ $B'' = 11$

$B' = 10 \quad B'' = 11$

B' $\quad$ B'

$(11) * 2^2 = 1100$

$+\ 01$

$$\begin{array}{r} 1100 \\ +0001 \\ \hline 1101 = A \end{array}$$

$$A = A' * 2^{n/2} + A''$$

$$B = B' * 2^{n/2} + B''$$

$$A \cdot B = (A' 2^{n/2} + A'') \cdot (B' 2^{n/2} + B'')$$

$$= A'B' \cdot 2^n + (A'B'' + A''B') 2^{n/2} + A''B''$$

$\underbrace{\text{4 multi of } n/2\text{-bit \#s}}$, $\underbrace{\text{2 shifts of } O(n)\text{-bit \#s} , \text{ 3 add of } O(n)\text{-bit \#s}}_{O(n)}$

Run Time: $T(n) = 4\,T(n/2) + O(n)$

$$\boxed{\text{Master Thm: } T(n) = a\,T(n/b) + n^d \Rightarrow O(n^{\log_b a}) \text{ if } d \le \log_b a}$$

$a = 4,\ b = 2,\ d = 1$

$$\Rightarrow O(n^{\log_2 4}) = O(n^2)\ !$$

no improvement :C

Clever idea: keep $\underbrace{A'B'}_{c_1}$, $\underbrace{A''B''}_{c_2}$

rewrite

$A \cdot B = A'B' \cdot 2^n + \boxed{A'B'' + A''B'}\, 2^{n/2} + A''B''$

$c_1 \quad c_2$

$+ \boxed{A'B'' + A''B'} \cdot 2^{\ldots}$

$+ A''B''$

rewrite

$$\underbrace{(A'+A'')\cdot(B'+B'')}_{c_3} = \cancel{A'B'} + \boxed{(A'B''+A''B')} + \cancel{A''B''}$$

$$- \underbrace{A'B'}_{c_1} - \underbrace{A''B''}_{c_2} \qquad - \cancel{A'B'} - \cancel{A''B''}$$

$$= c_1 \cdot 2^n + ((c_3 - c_1 - c_2)\cdot 2^{n/2} + c_2$$

Mult (A, B)

1. if n is cost ...
2. Divide A into A', A'' of $n/2$ bits each
   " B " B', B'' " "
3. $c_1 = \text{Mult}(A', B')$
   $c_2 = \text{Mult}(A'', B'')$
   $c_3 = \text{Mult}((A'+A'') \cdot (B'+B''))$
4. Return $\left(c_1 \cdot 2^n + ((c_3 - c_1 - c_2)\cdot 2^{n/2} + c_2\right)$

Run Time: $T(n) = 3\,T(n/2) + O(n)$

$$\Rightarrow \quad = O\left(n^{\lg_2 3}\right) = O\left(n^{1.59}\right)$$

Faster! :)

Rmk: can enhance idea.

$$T(n) = 5\,T(n/3) + O(n) = O(n^{1.47})$$

$$= 7\,T(n/4) + O(n) = O(n^{1.41})$$

$\vdots$

$= O(n^{1+\varepsilon})$ ε is a constant.
(Toom - Cook '63)

$\vdots$

$O(n \lg n \lg\lg n)$ (SS'71)

$\vdots$

$O(n \lg n \lg\lg\lg \cdots \lg n)$ (Fürer'07)

$\vdots$

$O(n \lg n)$ (CD'20).

Problem 2: Selection.

Given n unsorted numbers $a_1, \ldots, a_n$, $k \le n$ ← int.

Median: $k = \frac{n}{2}$

Find k^{th} smallest no.

eg. 50, 82, 45, 18, 90, 35, 75, 25

$k=4$, 45

Alg'm 1: Sort & return k^{th}

$O(n \lg n)$

Alg'm 2: Selection sort variant.

$O(k \cdot n)$

Alg'm 3: Heapsort variant

Alg'm 3: Heapsort variant

$$O(n + \underbrace{k \cdot \lg n}_{})$$

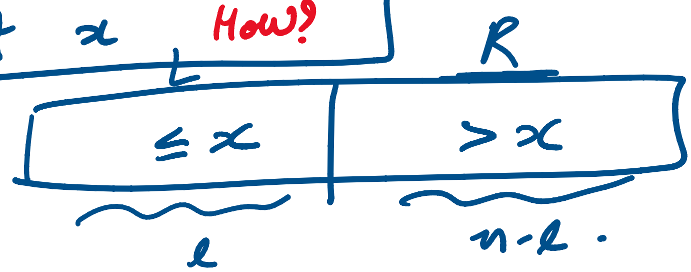
n ↑ build heap $k \lg n$ ↑ k-delete min.

Is $O(n)$ possible?

Alg'm 4: Quicksort variant.

$$\text{Select}(\{a_1, \ldots, a_n\}, k)$$

1. if $n=1$ return
2. Pick a pivot x How?



3. Construct $L = \{a_i \mid a_i \le x\}$ $|L| = l$

   $R = \{a_i \mid a_i > x\}$

   $O(n)$

4. if $(k \le l)$ then return $\underline{\text{select}(L, k)}$ $T(l)$

   else " $\underline{\text{select}(R, k-l)}$ $T(n-l)$

   $\max\{T(l), T(n-l)\}$

$$T(n) = \max\{T(l), T(n-l)\} + O(n)$$

ideal: $l = n/2$

$$T(n) = T(n/2) + O(n)$$

$$O(n + n/2 + \cdots + 1)$$

$$T(n) = T(n/2) \ldots$$

$$= O\left(n + \underline{n/2 + n/4 + \cdots + 1}\right)$$

$$= O(n + n) = O(n)$$

worst : $l = 1$ or $l = n-1$

$$T(n) = T(n-1) + O(n)$$

$$= O(n^2)$$

$$\boxed{\text{Can we get } O(n) \text{ ?}}$$

Alg'm by Blum, Floyd, Rivest, Pratt, Tarjan (1973)

Idea: Pick a pivot x close to the median.

By taking $\underbrace{\text{median}}_{\text{recursive call.}}$ of medians of 5

eg. $\underbrace{1, 10, 5, 8, 21}_{G_1}, \underbrace{34, 6, 7, 12, 23}_{G_2}, \underbrace{2, 4, 30, 11, 25}_{G_3}$

$x_1 = 8$ $\quad x_2 = 12$ $\quad x_3 = 11$

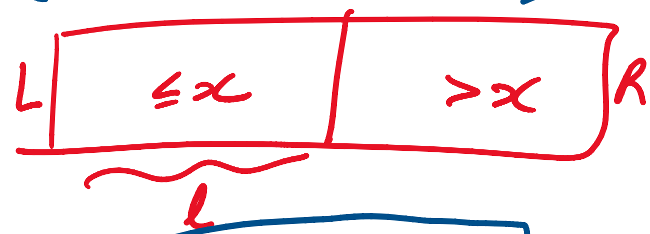
$x = 11$ pivot.

Replace line 2:

2.1 split $\{a_1, \ldots, a_n\}$ into groups $G_1 \ldots G_{n/5}$ each of 5 #s.

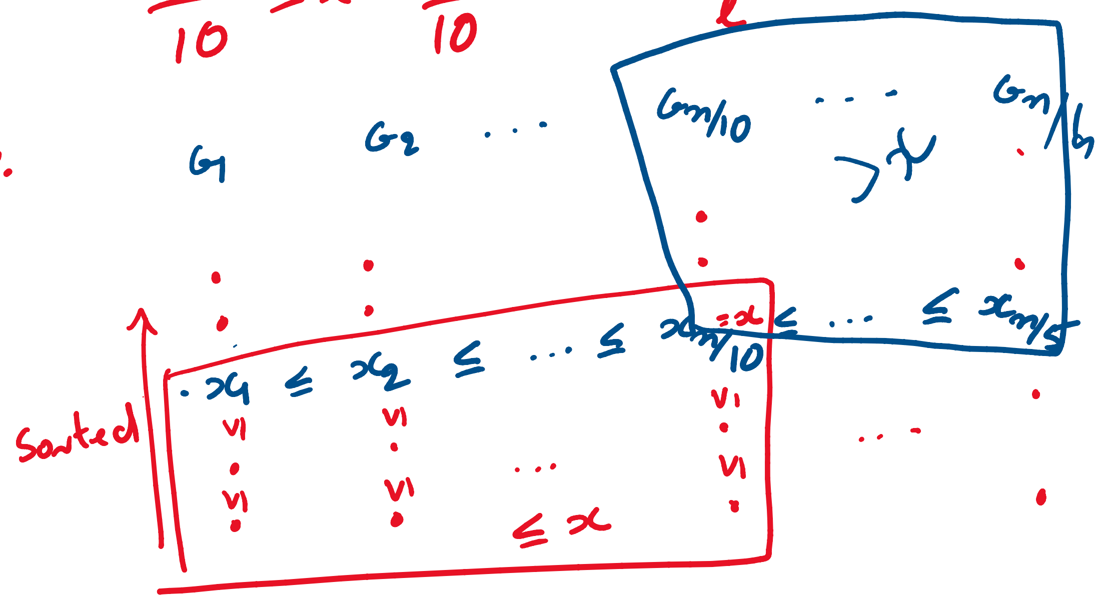
2.2. for $i = 1$ to $n/5$ x_i = median of G_i

2.3. $x = \text{Select}\left(\{x_1, x_2, \ldots, x_{n/5}\}, \frac{n}{10}\right)$



Lemma: $\frac{3n}{10} \leq \ell \leq \frac{7n}{10}$

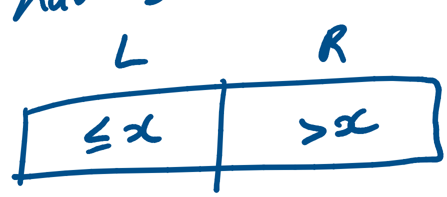
pf:



$n/10$ groups have their median $x_i \leq x$.

Each of these groups have 3 #s $\leq x_i \leq x$

$\Rightarrow$ $|L| = \ell \geq \frac{3n}{10}$



By symmetric argument.

$|R| = (n - \ell) \geq \frac{3n}{10} \Rightarrow \ell \leq \frac{7n}{10}$ ∎

$$|R| = (n-l) \geq \frac{3n}{10} \implies l \leq \frac{7n}{10} \quad \blacksquare$$

Run time:

$$T(n) = T\left(\frac{7n}{10}\right) + T\left(\frac{n}{5}\right) + O(n)$$

$$= O(n)$$

Guess & Verify $\quad \left(\because \frac{7}{10} + \frac{1}{5} = \frac{9}{10} < 1\right)$