Thursday, February 24, 2022 10:46 AM

Divide & Conquer

Problem 1: Multiplying Large Numbers.

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

$B = b_{n-1}\, b_{n-2} \ldots b_0$

Compute $AB = C = c_{2n-1}\, c_{2n-2} \ldots c_0$

"Elementary school" Algo: ('' B.C.)

$$
\begin{array}{r}
A = 1101 = 13 \\
B = 1011 = 11 \\
\hline
1101 \\
1101 \\
1101 \\
\hline
10001111
\end{array}
$$

($O(n)$ shifts

+ $O(n)$ additions)

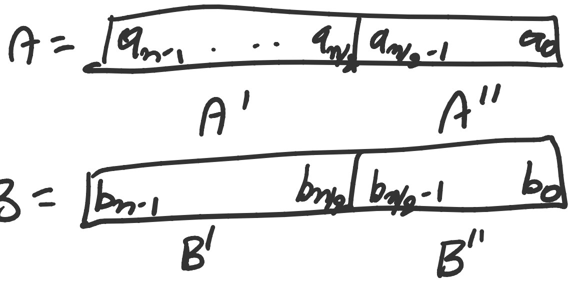
+ $\Theta(n)$

Total time: $\Theta(n^2)$

Surprisingly we can do better!

✯ Karatsuba's Algo (1962):

Approach: Divide



$A = 1101 = 13$

$A' = 11 = 3$

$A'' = 01 = 1$

$$
\begin{array}{r}
1100 = A' * 2^2 \\
+ \quad 01 \\
\hline
1101
\end{array}
$$

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

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

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

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

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

$(\underbrace{\text{3-addition, 2-shifts,}}_{O(n)}$ 4 Multiplications on $n/2$ bit numbers$)$

$$\Rightarrow T(n) = 4\,T\left(\frac{n}{2}\right) + O(n)$$

["Master" Thm: $T(n) = a\,T\left(\frac{n}{b}\right) + O(n^d)$, $d < \frac{a}{b}$

Then $O(n^{\log_b a})$]

$a=4, b=2$

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

NO Progress!

Clever idea:

$$(A'+A'')\cdot(B'+B'') = A'B' + \underbrace{A'B'' + A''B'} + A''B''$$

$$\Rightarrow \underline{A'B'' + A''B'} = \underline{(A'+A'')\cdot(B'+B'') - A'B' - A''B''}$$

(6 addition, 2 shifts, 3 multiplications)

$$\underline{A'B'},\ A''B'',\ (A'+A'')\cdot(B'+B'')$$

Mult(A, B)

1. if $n=1$ then constant work
2. Else divide A into A', A'' of $n/2$ bits each
   " B " B', B'' " "

$$\text{Mult.}^{\frac{n}{2}\text{ bit}} \left\{ \begin{array}{l} 3.\quad C_1 = \text{Mult}(A', B') \\ 4.\quad C_2 = \text{Mult}(A'', B'') \\ 5.\quad C_3 = \text{Mult}\big((A' \pm A''),\ (B' \pm B'')\big) \end{array} \right.$$

6. Return : $C_1 2^n + (C_3 - C_1 - C_2) \cdot 2^{n/2} + C_3$

$$\Rightarrow T(n) = 3\,T\left(\frac{n}{2}\right) + O(n)$$

$$\Rightarrow O\left(n^{\log_2 3}\right) = O\left(n^{1.59}\right)$$

Significant Progress !

Note : Can be improved

$$T(n) = 5\,T\left(\frac{n}{3}\right) + O(n) \rightarrow O\left(n^{1.47}\right)$$

$$T(n) = 7\,T\left(\frac{n}{4}\right) + O(n) \rightarrow O\left(n^{1.41}\right)$$

$$\vdots$$

$O(n^{1+\varepsilon})$ for any constant $\varepsilon > 0$.
(Toom-Cook '63)
″

$\rightarrow O(n \log n \log\log n)$ (Scöhage - Strassen '71)

$O(n \log n \underbrace{\log\log\log \cdots \log}\, n)$ (Furer '07)

$O(n \log n \underbrace{\log\log\log \cdots \log n}_{\text{constantly many}})$ (Furer

$O(n \log n)$ (Harvey - Hoeven '20)

Is this the best? OPEN!!

Problem 2: Selection.

Given n numbers (unsorted) $a_1, a_2, \ldots, a_n$

Find k^{th} smallest number.

eg. 60, 80, 42, 10, 99, 75, 35, 25

$k=4 \Rightarrow 42$

e.g. $k = \lfloor \frac{n}{2} \rfloor$

Median.

Algo 0: Sort & look up

$O(n \lg n)$ time

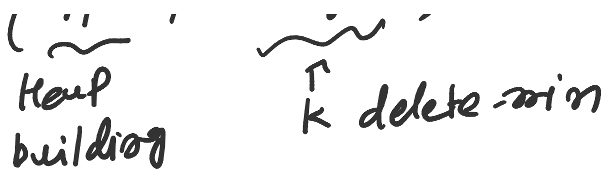
Algo 1: Selectionsort Variant

$O(kn)$ time ←

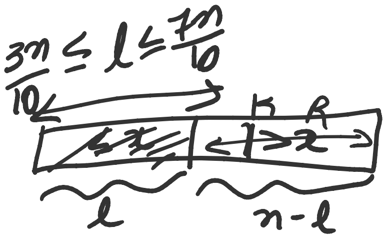
Algo 2: Heapsort Variant

$(\underbrace{n}_{\text{Heap}} + \underbrace{k \lg n})$ if $k = \frac{n}{\lg n} \Rightarrow O(n)$

↑ delete-min



Algo 3: Quicksort Variant.



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

1. If $n=1$ return a_1
2. Pick a pivot x How?
3. Partition into $L = \{a_i \mid a_i \leq x\}$ $l = |L|$

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

$O(n)$ $\}$

4. If $(l \geq k)$ then return select(L, k)

   Else return select$(R, k-l)$

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

Best case: $l = \frac{n}{2}$

$$T(n) = T\left(\frac{n}{2}\right) + O(n)$$

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

$$= O(n)$$

Worst case $l=1$ OR $l=(n-1)$

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

$$= O\left(n + (n-1) + (n-2) + \cdots + 1\right)$$

$= O(n^2)$

Algo by Blum, Floyd, Rivest, Pratt, Tarjan (1973)

Clever idea: Pick a good pivot x close to median by taking median of medians of 5s.

Replace Line 2 by: $\{a_1, \ldots a_n\}$

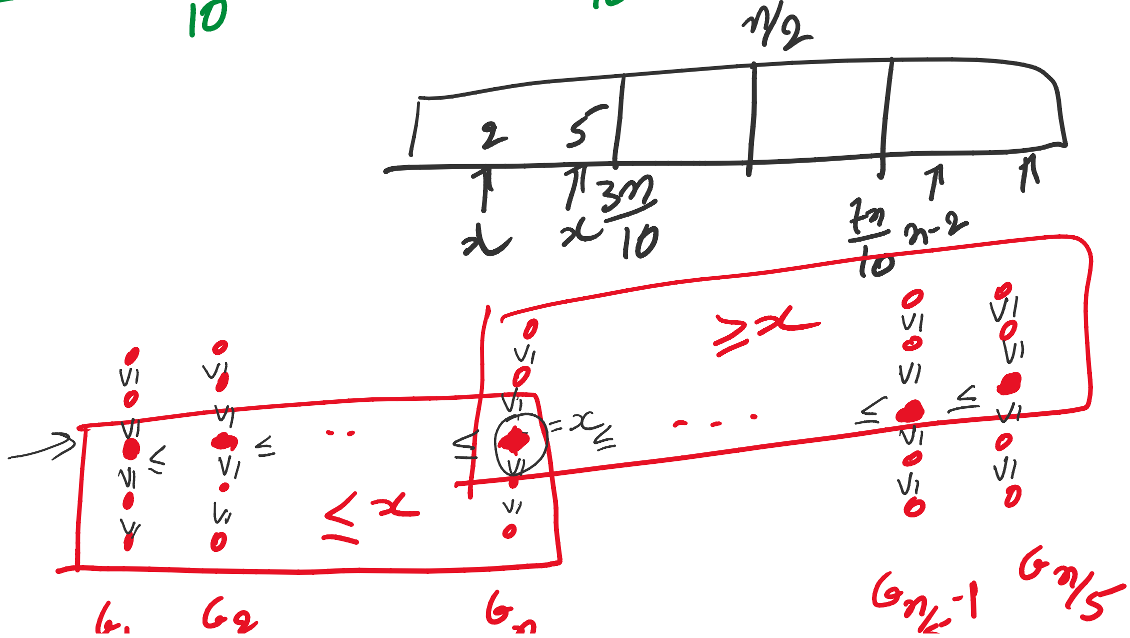
2.1 Split $\{a_1, \ldots a_n\}$ into groups $G_1, \ldots, G_{n/5}$ each of 5 ele.

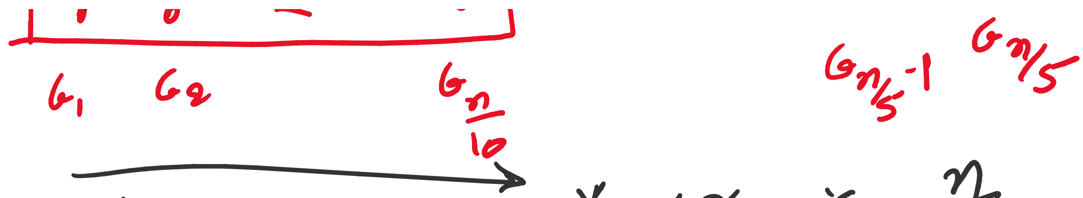
$O(n) \Rightarrow$ 2.2 For $i = 1$ to $n/5$ do $X_i =$ median of G_i

2.3. $x = \text{select}(\underline{\{X_1, \ldots, X_{n/5}\}}, \frac{n}{10})$

Lemma: $\frac{3n}{10} \le l \le \frac{7n}{10}$

Pf:





→ # groups w/ $X_i \le x$ is $\frac{n}{10}$

→ In each group $\underline{3}$ elements $\le X_i \le x$

& these $\frac{n}{10}$

→ Atleast $\frac{3n}{10}$ elements $\le x$.

By symmetric argument $\frac{3n}{10} \ge x$

$$\Downarrow$$

$$\frac{3n}{10} \le \ell \le \frac{7n}{10}$$

$$T(n) = \max\left\{T(\ell),\ T(n-\ell)\right\} + T\left(\frac{n}{5}\right) + O(n)$$

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

(Guess & Verify)

$$= \underline{O}(n)$$

$$\left(\because \quad \frac{7}{10} + \frac{1}{5} < 1\right)$$