Lecture 12 Scribble

Chat Moderators: Junyeob
Eliot

Topics: - Kartsuba's Algorithm
- Quick Select
- Median of Medians

Kartsuba's algorithm divide & conquer multiplying numbers

How do we multiply 2 numbers

Algorithm 0: Lookup Table

Algorithm 1: $b \cdot c$

for ($i = 1 \to c$) $b += b$:

Algorithm #2: Egyptian Algorithm

- break the multiplier into base 2 component and then add

b	c	Sum
76	35	
76	34+1	
76	34	76
152	17	
152	16+1	
152	16	152
304	8	
608	4	
1216	2	
2432	1	2432

2660 $= 76 \cdot 35$

Algorithm #3 : Two n digits #'s   Westerm Multiplicatio

$$O(n^2)$$

```
      2130
   x  4001
   -------
      2130
     0000
    0000
   8570
----------
  8522130
```

Algorithm #4 : Divide & Conquer   b & c have n digits

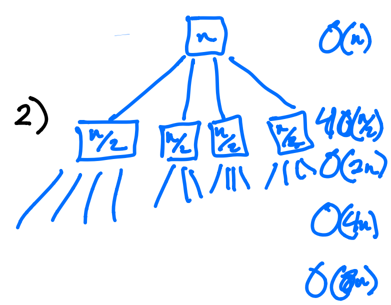
break b & c into 2 halves

$$b = b_L \cdot 2^{n/2} + b_R \qquad c = c_L \cdot 2^{n/2} + c_R$$

$$b \cdot c = b_L \cdot c_L \cdot \underbrace{2^{n/2} \cdot 2^{n/2}}_{\equiv \, << n} + b_R c_L \cdot \underbrace{2^{n/2}}_{\equiv \, << n/2} + b_L c_R \cdot 2^{n/2} + b_R c_R$$

```
mult4 (b[1...n], c[1...n])
    if len(b)==1 : return b[0] & c[0]
    bLcL = mult4 (b[n ... n/2], c[n ... n/2]) << (n/2 • 2)
    bRcL = mult4 (b[n/2 ...1], c[n ... n/2]) << n/2
    bLcR = mult4 (b[n ... n/2], c[n/2 ... 1]) << n/2
    bRcR = mult4 (b[n/2 ... 1], c[n/2 ... 1])
    return bLcL + bRcL + bLcR + bRcR
```

$$T(n) = 4T(n/2) + O(n)$$



$$\#\text{leaves} \cdot O(1) = r^{\log_2 n} =$$

$$T(n) = O(n^{\log_2 r}) = O(n^2)$$

$$T(n) = rT(n/c) + f(n)$$

Section 1.7
Algorithms by
Jeff Erickson

Algorithm 5: Karatsuba's Algorithm

Gauss's observation

$$(a+bi)(c+di) = ac - bd + (ad+bc)i$$

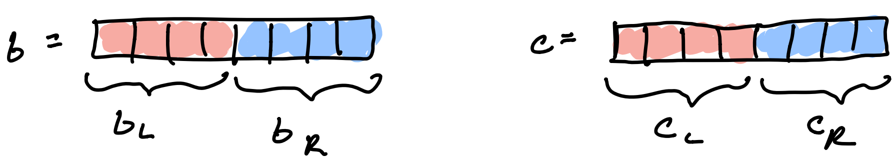
$$(a+b)(c+d) = ac + bd + ad + bc$$

$$(ad+bc) = (a+b)(c+d) - ac - bd$$

$$p(x) = ax+b \qquad q(x) = cx+d$$

$$p(x)q(x) = ac\,x^2 + (ad+bc)x + bc$$

$$p(x)q(x) = ac\,x^2 + \big((a+b)(c+d) - ac - bd\big)x + bd$$



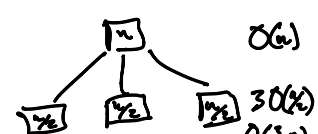
$$b = b_L \cdot 2^{n/2} + b_R \qquad c = c_L \cdot 2^{n/2} + c_R$$

$$b \cdot c = b_L \cdot c_L \cdot 2^{n/2} \cdot 2^{n/2} + b_R c_L \cdot 2^{n/2} + b_L c_R \cdot 2^{n/2} + b_R c_R$$

$$= b_L c_L \cdot 2^n + \big((b_L + c_L)(b_R + c_R) - b_L c_L - b_R c_R\big) \cdot x + c_R b_R$$

```
mult4(b[1...n], c[1...n])
    if len(b)==1 : return b[0] & c[0]
    bLcL = mult4(b[n...n/2], c[n...n/2]) << (n/2 • 2)
    bRcL = (mult4(b[n/2...1] + c[n/2...1], b[n...n/2] + c[n...n/2])
             - bLcL - bRcR) << n/2
    bRcR = mult4(b[n/2...1], c[n/2...1])
    return bLcL + bRcL + bLcR + bRcR
```

$$T(n) = 3T(n/2) + O(n)$$



$$T(n) = O(n^{\log_2 3})$$

$$= O(n^{1.585})$$

Selection in linear time

Unsorted array of n ints

We want to find element of rank k

16	96	8	35	43	42	4	13	23
4	9	2	6	8	7	1	3	5

Algorithm #1 Sort, then look at A[k] ⇒ $O(n \log n)$

Algorithm #2: Run k for minimum k time $O(kn)$

if $k = n/2$ $= O(n^2)$

Algorithm #3 QuickSelect

n

16	96	8	35	43	42	4	13	23
4	9	2	6	8	7	1	3	5

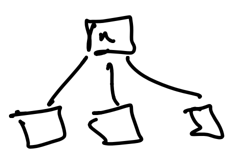
$\Downarrow$ $O(n)$

8	4	13	16	96	35	43	42	23

4 x.

$T(n) \leq T(x) + O(n)$ Assume $\frac{1}{4}n \leq x \leq \frac{3}{4}n$

$T(n) = T(\frac{3}{4}n) + O(n)$



$O(n)$

$O(\frac{3}{4}n)$

$O(\frac{9}{16}n)$

$T(n) = O(n)$

Median of Medians

$\frac{3}{10}n - 6 \leq x \leq \frac{7}{10}n + 6$

$T(n) = T(\lceil \frac{n}{5} \rceil) + O(1)$ $\frac{n}{5} O(1)$

$T(n) = T(\lceil \frac{n}{5} \rceil) + T(\max[|A_<|, |A_>|]) + O(n)$

$T(n) \quad T(\frac{1}{5} n) + T(\frac{7}{10} n) + O(n) \longrightarrow O(n)$

sort 5 element array ↓ $\quad$ $O(1)$ sort 5 elements

$\frac{n}{5} O(1)$

$M(n) = \frac{n}{5} O(1)$
$= O(\frac{n}{5})$

Find Rank k $(A[1 \ldots n], k)$ $\quad T(n)$

Find Medians of M

- Sort 5 element arrays $= M[1 \ldots \frac{n}{5}]$ $\quad O(1)$

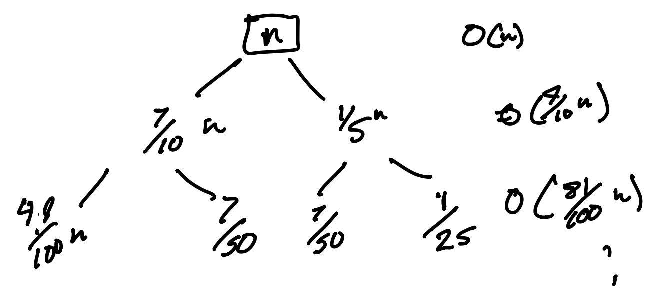
FindRankk$(M[1 \ldots \frac{n}{5}], 0.5])$ //Gives us pivot $\quad T(\frac{n}{5})$

Using pivot break into Left & Right $\quad O(n)$

FindRankk$(A_<$ or $A_>, k)$ $\quad T(\max[|A_<|, |A_>|])$

$$T(n) = T(\max(|A_<|, |A_>|)) + T(\tfrac{1}{5} n) + O(n) + O(1)$$

$$T(n) = T(\tfrac{7}{10} n) + T(\tfrac{1}{5} n) + O(n)$$



49 + 14 + 14 + 4

$T(n) = O(n)$

Lecture 12 Addendum Scribble - Quick Select

16	96	8	35	43	42	4	13	23
4	9	2	6	8	7	1	3	5

Find the integer of rank k in an unsorted array of n elements

Algorithm 1: Sort the array. Return $A[k]$ $O(n \log n)$

Algorithm 2: Find k minimum elements in A $O(kn)$

- Smallest element in A
- Remove that element
- Repeat k times

If $k = n/2$ then $O(\frac{n}{2} \cdot n) = O(n^2)$

Algorithm 3: Quick Select

16	96	8	35	43	42	4	13	23

8	4	13	16	96	35	43	42	23
			4			7		
	$A_<$					$A_>$		

Quick Select $(A[1 \ldots n], k)$

// Choose Pivot

Pivot $= A[1]$ $O(1)$

$A_<, A_> =$ partitioning$(A[1 \ldots n]$, pivot$)$ $O(n)$

If rank(pivot) $== k$

return pivot

else

if rank(pivot) > k
    return QuickSelect ($A_<$, k) $T(\cancel{\text{rank(pivot)}}) = |A_<|$
else
    return QuickSelect ($A_>$, k - rank(pivot)) $T(\cancel{n - \text{rank pivot}}) = |A_>|$

$$T(n) = T(\max[|A_<|, |A_>|]) + O(n) + O(1)$$

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

$$T(n) = T(n-k) + O(n) \approx O(n^2)$$
where k is small

Need a way to choose pivot such that rank(pivot) near the middle.

Median of Medians: We can choose a pivot (p) $\frac{3}{10}n - 6 \le p \le \frac{7}{10} + 6$

QuickSelect (A[1...n], k)

\\ Choose Pivot

Break A into n/5 - 5-element arrays $O(1)$

Finding the median of each 5-element arrays $O(n)$

pivot = QuickSelect ($A_{median}[1 \dots n/5]$, n/10) $T(\frac{1}{5}n)$

$A_<, A_>$ = partitioning (A[1...n], pivot) $O(n)$

If rank(pivot) == k
    return pivot
else
    if rank(pivot) > k
        return QuickSelect ($A_<$, k)
    else
        return QuickSelect ($A_>$, k)

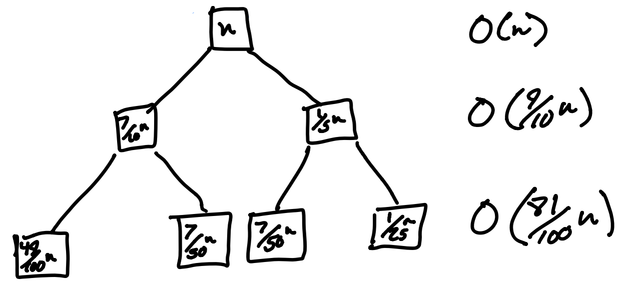
$T(\max[|A_<|, |A_>|])$

$A_<, A_> \le \frac{7}{10}n + 6$

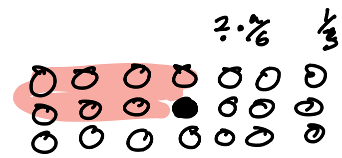
$T(\frac{7}{10}n + 6) \approx T(\frac{7}{10}n)$

$$T(n) = T(\tfrac{7}{10}n) + T(\tfrac{1}{5}n) + O(n) + O(n) + O(1)$$

$$T(n) = T(\tfrac{7}{10}n) + T(\tfrac{1}{5}n) + O(n)$$

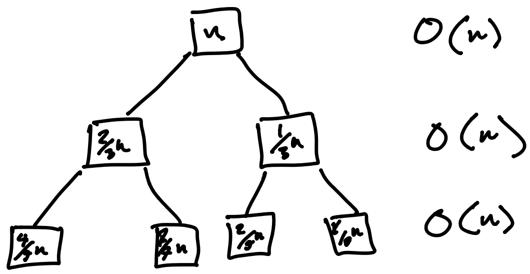


$T(n) = O(n)$



Why not 3-element subarrays

$T(n) = T(\frac{2}{3}n) + T(\frac{1}{3}n) + O(n) = O(n \log n)$



What about 10 element subarrays?

$T(n) = T(\frac{13}{18}n) + T(\frac{1}{9}n) + O(n) = O(n)$