CSE 201 2-6-20

Sec. 9.2: The Selection Problem

Defn

Let $A[1....n]$ contain distinct elements. The i^{th} order statistic is the i^{th} smallest element. i.e. if $A[1...n]$ were sorted, its the element $A[i]$. i.e. the i^{th} ord. stat. is the unique element that is greater than $i-1$ other elements.

Selection Problem:

given $A[1 \ldots n]$ (distinct elements)
and i in range $1 \le i \le n$,
find the i^{th} ord. statistic

$i = 1$: minimum
⋮
$i = n$: maximum

obvious solution

- sort $A[1 \ldots n]$
- return $A[i]$

cost $= \Theta(n \log n)$

<u>Recall</u>: RandPartition(A, p, r) returns

q in range $p \leq q \leq r$ and

causes

$$A[p \cdots q-1] \underset{<}{\leq} A[q] < A[(q+1) \cdots r]$$

also q is equally likely to be any of $p, p+1, p+2, \ldots, r$. Thus

Thus

$$P(q=j) = \frac{1}{r-p+1} \quad (\text{for } p \leq j \leq r)$$

RandSelect(A, p, r, i) Pre: $1 \le i \le r-p+1$

```
if p = r
    return A[p]
q = RandPartition(A, p, r)
k = q - p + 1        // length(A[p...q])
if k = i
    return A[q]
else if k > i
    return RandSelect(A, p, q-1, i)
else    // k < i
    return RandSelect(A, q+1, r, i-k)
```

<u>Rmk</u> Rand-
much like Quicksort, also
much like Binary Search

Let $t(n)$ = average case # of
array comparisons by:
RandSelect$(A, 1, n, i)$

so

$$t(n) = \frac{\sum_{q=1}^{n}\Big((n-1) + P(q>i)\,t(q-1) + P(q<i)\,t(n-q)\Big)}{n}$$

where
$$P(q>i) = P(i<q\leq n) = \frac{n-i}{n}$$
$$P(q<i) = P(1\leq q<i) = \frac{i-1}{n}$$

So

$$t(n) = (n-1) + \frac{1}{n} \cdot \sum_{q=1}^{n} \left(\left(\frac{n-i}{n}\right) t(q-1) + \left(\frac{i-1}{n}\right) t(n-q) \right)$$

$$= (n-1) + \frac{1}{n^2} \cdot \left[(n-i) \cdot \sum_{q=\cancel{1}2}^{n} t(q-1) + (i-1) \cdot \sum_{q=1}^{\cancel{n}\, n-1} t(n-q) \right]$$

$$= (n-1) + \frac{1}{n^2} \left[(n-i) \cdot \sum_{q=1}^{n-1} t(q) + (i-1) \cdot \sum_{q=1}^{n-1} t(q) \right]$$

$$= (n-1) + \frac{1}{n^2} (n - \cancel{i} + \cancel{i} - 1) \cdot \sum_{q=1}^{n-1} t(q)$$

and hence

$$t(n) = (n-1) + \left(\frac{n-1}{n^2}\right) \cdot \sum_{q=1}^{n-1} t(q)$$

See Prob #3 (?) on mid 1

review soln, where its shown

$$\forall n \geq 1 : \; t(n) \leq 2n$$

$\therefore \; t(n) = O(n)$

Also $t(n) \geq n-1 = \Omega(n)$

$\therefore \; t(n) = \Omega(n)$

$\therefore \; t(n) = \Theta(n)$.

Exercise (hard)

find exact solution $t(n) = ???$

Exercise

- write non-randomized version
- Prove correctness of this version (non-random.)
- write a recurrence for worst case # comp.
- show $T(n) = \Theta(n^2)$

Sec. 4.2 (P.75-83)

Strassen's algorithm

<u>Problem</u> given two $n \times n$ matrices A and B, find Product

$$C = A \cdot B$$

another $n \times n$ matrix.

Recall: if $C = (c_{ij})$, $A = (a_{ij})$, $B = (b_{ij})$ for $1 \le i \le n$, $1 \le j \le n$, then

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

Basic op.: floating point mult.
(book includes addition)

cost $= \Theta(n^3)$

Assume n is an exact power of 2. $\therefore$ so are $n/2, n/2^2, n/2^3, \ldots$

<u>Divide</u> A, B into sub-matrices

$$A = \left(\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right), \quad B = \left(\begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array}\right)$$

$\frac{n}{2} \times \frac{n}{2}$

Solve: Perform 8 multiplications of $\frac{n}{2} \times \frac{n}{2}$ submatrices

$$C_{11} = A_{11} B_{11} + A_{12} B_{21}$$

$$C_{12} = A_{11} B_{12} + A_{12} B_{22}$$

$$C_{21} = A_{21} B_{11} + A_{22} B_{21}$$

$$C_{22} = A_{21} B_{12} + A_{22} B_{22}$$

Combine:

$$C = \left(\begin{array}{c|c} C_{11} & C_{12} \\ \hline C_{21} & C_{22} \end{array}\right)$$

Exercise:

- this Process correctly computes $C = A \cdot B$.
- write Pseudo-code. (p. 77 of text.)

Let $T(n)$ = # of Basic OPs. by this recursive Procedure.

observe

$$T(n) = \begin{cases} 1 & n = 1 \\ 8T\left(\frac{n}{2}\right) + 1 & n \geq 2 \text{ (exact Pow. of 2.)} \end{cases}$$

cost of dividing & combining
book has n^2 since they count additions also

By master thm:

Compare $1 = n^0$ to $n^{\log_2 8} = n^3$

by case 1: $T(n) = \Theta(n^3)$.

1969: Volker Strassen found a way to do this with only 7 multiplications of $\frac{n}{2} \times \frac{n}{2}$ matrices.

Details p. 80-81.

Runtime

$$T(n) = 7T\left(\frac{n}{2}\right) + 1$$

compare: $1 = n^0$ to $n^{\log_2 7} = n^{2.8073\ldots}$

case 1: $T(n) = \Theta(n^{\lg(7)})$.

<u>Exercise</u>

Let N be smallest Pow. of 2 that is greater than or equal to n. Show that

$$N^{\lg(7)} = \Theta(n^{\lg(7)})$$

<u>Exercise</u>

Do same for non-square matrices $n \times m$, i.e. find a fcn $N(n,m)$ which is an exact pow. of 2, and find $\Theta(N(n,m)^{\lg(7)}) = ?$

Dynamic Programming (sec. 15.1-15.5)

Problem: Binomial coefficients

Recall

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!}$$

better (Pascale's identity)

$$\binom{n}{k} = \begin{cases} 1 & \text{if } k=0,\ k=n \\ \binom{n-1}{k-1} + \binom{n-1}{k} & \text{if } 0<k<n \end{cases}$$

BinCoef(n, k) Pre: $0 \le k \le n$

1. if $k = 0$ or $k = n$
2.   return 1
3. else
4.   return BinCoef(n-1, k-1) + BinCoef(n-1, k)

Recall: if $n = 2k$

$$\binom{n}{k} = \binom{2k}{k} = \Theta\left(\frac{4^k}{\sqrt{k}}\right)$$