cse 102 2-6-20

chapter 9: the Selection Problem

another (related) Problem:

given A[1....n], find both min and max elements, and return as a pair: (min, max)

Subroutines:

min(x,y)

1. return (x<y? x:y)

1 comparison

max(x,y)

1. return (x>y? x:y)

1 comparison

MinMax(A, p, r) Pre: $p \le r$

```
if p = r
    return (A[p], A[p])
else
    q = ⌊(p+r)/2⌋
    (m1, M1) = MinMax(A, p, q)
    (m2, M2) = MinMax(A, q+1, r)
    return (min(m1, m2), max(M1, M2))
```

let $T(n)$ = # of array comparisons by MinMax$(A, 1, n)$.

so

$$T(n) = \begin{cases} 0 & n = 1 \checkmark \\ T\left(\left\lceil \frac{n}{2} \right\rceil\right) + T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + 2 & n \geq 2 \checkmark \end{cases}$$

Recall: if $q = \left\lfloor \frac{1+n}{2} \right\rfloor$

$$\text{length } A[1 \cdots q] = q - 1 + 1 = q = \left\lfloor \frac{1+n}{2} \right\rfloor = \left\lceil \frac{n}{2} \right\rceil$$

exercise

$$\text{length } [q+1 \cdots r] = \cdots\cdots = \left\lfloor \frac{n}{2} \right\rfloor$$

<u>exercise:</u> use master thm

first simplify: $T(n) = 2T\left(\frac{n}{2}\right) + 1$

answer: $T(n) = \Theta(n)$

<u>check</u>: $T(n) = 2n-2$ is exact solution!

$$\begin{aligned}
RHS &= T\left(\left\lceil \tfrac{n}{2} \right\rceil\right) + T\left(\left\lfloor \tfrac{n}{2} \right\rfloor\right) + 2 \\
&= \left(2\left\lceil \tfrac{n}{2} \right\rceil - 2\right) + \left(2\left\lfloor \tfrac{n}{2} \right\rfloor - \cancel{2}\right) + \cancel{2} \\
&= 2\left(\left\lceil \tfrac{n}{2} \right\rceil + \left\lfloor \tfrac{n}{2} \right\rfloor\right) - 2 \\
&= 2n - 2 \\
&= T(n) = LHS.
\end{aligned}$$

<u>Exercise</u> Read 9.1 description of iterative algorithm (3 comp/2 elements)

(a) write iterative version

(b) write recursive version

(c) write a recurrence for (b)

(d) show exact soln. to (c) is: $\left\lceil \frac{3n}{2} \right\rceil - 2$

Selection Problem

Defn

given an array $A[1 \cdots n]$ of distinct elements, the i^{th} order statistic is the i^{th} smallest element. i.e. the unique element that is greater than $i-1$ other elements.

For instance

$i = 1$: minimum

$\vdots$

$i = n$: maximum

i.e. if $A[1 \cdots n]$ were sorted, the i^{th} ord. stat. is element belonging at $A[i]$

<u>Problem</u> (selection)
given $A[1 \cdots n]$ distinct, and $1 \le i \le n$, find i^{th} order stat.

<u>obvious solution</u>

sort $A[1 \cdots n]$, then return $A[i]$.

Cost: $\Theta(n \log n)$

RandSelect is a randomized algorithm that solves in (average case) $\Theta(n)$ time.

Recall RandPartition(A, p, r) causes

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

$<$
since
distinct
elements

Pre: $1 \le i \le r-p+1$

RandSelect(A, p, r, i)

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

<u>Rmks</u>

(1) a lot like Quicksort

(2) a lot like Binary search
only 1 recursive call

<u>Exercise</u>

Prove correctness.

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

where $1 \le i \le n$.

Recall: RandPartition $(A, 1, n, i)$ does $n-1$ comparisons. Also q is equally likely to be any of $1, \ldots, n$; i.e.

$$P(q = \text{anything}) = \frac{1}{n}$$

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

where

$$P(i < q \le n) = \frac{n-i}{n}$$

$$P(1 \le q < i) = \frac{i-1}{n}$$

$(i-1) - \cancel{1} + \cancel{1} = i-1$

Thus

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

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

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

So

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

<u>Exercise</u> Prove $t(n) = \Theta(n)$.

<u>Exercise</u> (hard) find exact solution.

Strassen's Algorithm (sec. 4.2 p. 75-83)

Problem multiply two $n \times n$ matrices: A, B

$$C = A \cdot B$$

defn

$$C_{ij} = \sum_{k=1}^{n} A_{ik} \cdot B_{kj} \quad \begin{cases} 1 \le i \le n \\ 1 \le j \le n \end{cases}$$

Basic operation:

- multiplication of 2 real numbers.
- (book includes addition ops.)

cost = $\Theta(n^3)$

what if n is an exact power of 2.

Recursive Procedure

divide:

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

$$A = \underbrace{\left(\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right)}_{n} \; n$$

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

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

solve

$$C_{11} = A_{11}B_{11} + A_{12}B_{21} \quad \Big\{ \tfrac{n}{2} \times \tfrac{n}{2}$$

$$C_{12} = A_{11}B_{12} + A_{12}B_{22} \quad \Big\{ \text{"}$$

$$C_{21} = A_{21}B_{11} + A_{22}B_{21} \quad \Big\{ \text{"}$$

$$C_{22} = A_{21}B_{12} + A_{22}B_{22} \quad \Big\{ \text{"}$$

Combine

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