TCS-503: Design and Analysis of Algorithms

Median and Order Statistics

Unit I: Syllabus

- **Introduction:**
  - Algorithms
  - Analysis of Algorithms
  - Growth of Functions
  - Master's Theorem
  - Designing of Algorithms

Unit I: Syllabus

- Sorting and Order Statistics
  - Heap Sort
  - Quick Sort
  - Sorting in Linear Time
    - Counting Sort
    - Bucket Sort
    - Radix Sort
  - Medians and Order Statistics

What is Median?

The median of a set of numbers

is the number such that

half of the numbers are larger

and

half smaller.

A = [50, 12, 1, 97 , $\boxed{30}$]

How might we calculate the median of a set?

Sort the number and then pick (n/2) element.

A = [1, 12, $\boxed{30}$, 50, 97]

Run Time?

Using Mergesort or Heapsort

$\Theta(n \log n)$

Order Statistics

The i^{th} order statistic

in a set of n elements

is the i^{th} smallest element.

3	4	13	14	23	27	41	54	65

The minimum of a set of elements:

$i=1$

1^{st} order Statistics

Median

$i=n/2$ if n is odd

$(n/2)^{th}$ OS

The maximum of a set of elements:

$i=n$

n^{th} order statistics

Order Statistics

3	4	13	14	23	27	41	54	65	75

8^{th} order statistic

3	4	13	14	23	27	41	54	65	75

lower median upper median

$i = \lfloor (n+1)/2 \rfloor = n/2$ (lower median) and $\lceil (n+1)/2 \rceil = n/2+1$ (upper median), when n is even

$\lfloor \ \rfloor$ is the least integer function

$\lceil \ \rceil$ is the greatest integer function

Order Statistics

How to find the i^{th} element?

Naïve algorithm?

Sort array A, and find the element A[i].

If we use merge sort or randomized quicksort:

Worst-case running time $T(n) = \Theta(n\lg n)$

Can we do better than that?

Related with sorting, but different.

Our expected time is $\Theta(n)$.

Order Statistics

Finding Minimum or Maximum

Can we find max and min in $\Theta(n)$ time?

Let's See!!!!

Alg.: MINIMUM(A, n)

```
min ← A[1]
for i ← 2 to n
    do if min > A[i]
        then min ← A[i]
return min
```

i n=9

A	13	4	3	14	23	27	41	54	65
	1	2	3	4	5	6	7	8	9

Order Statistics

Finding Minimum or Maximum

Alg.: MINIMUM(A, n)

```
1 min ← A[1]
2 for i ← 2 to n do
3     if min > A[i] then
4         min ← A[i]
5 return min
```

i   n=9

A	13	4	3	14	23	27	41	54	65
	1	2	3	4	5	6	7	8	9

Line 1: min=A[1] =13

Line 2: will be executed from i=2 to 9.

Line 3: 13> 4 :Yes, then

Line 4: min=4

Line 2: i=3

Line 3: 4> 3 : Yes, then

Line 4: min=3

Line 2: i=4

Line 3: 3>14: No :

Line 4 will not be executed.

……………………….

Order Statistics

Finding Minimum or Maximum

Finding Minimum:

How many comparisons are needed?

$n - 1$

Finding Maximum:

The same number of comparisons are needed to find the maximum:

$n - 1$

Find min and max independently:

Total of $2n - 2$ comparisons

Order Statistics

Simultaneous Min, Max

Case 1: n is odd, say 5. $A = \{2, 7, 1, 3, 4\}$

1. *Set both min and max to the first element:*

   Set min = max = 2

2. Compare rest of the elements in pairs:

   Compare the larger element to the maximum so far, and compare the smaller element to the minimum so far.

   Compare 7 with 1, $\Rightarrow 1 < 7$

   $\Rightarrow$ Compare 1 with min and 7 with max

   3 comparisons

   $\Rightarrow$ min =1 and max=7

Order Statistics

Simultaneous Min, Max

$A = \{2, 7, 1, 3, 4\}$

Compare 3 with 4 $\Rightarrow 3 < 4$

$\Rightarrow$ compare 3 with min and 4 with max

3 comparisons

$\Rightarrow$ min = 1, max = 7

We performed= 6 comparisons

$= 3(5-1)/2$

$=3(n-1)/2$

Order Statistics

Simultaneous Min, Max

Case 2: n is even, say 6. A = {2, 5, 3, 7, 1, 4}

1. *Perform 1 comparison on the first two elements to determine the initial values of the minimum and maximum:*

   Compare 2 with 5: $\Rightarrow$ 2 < 5

   Set min = 2, max = 5

   1 comparison

2. *process the rest of the elements in pairs as in the case for the odd one i.e. compare the first two elements, assign the smallest one to min and the largest one to max*

Order Statistics

Simultaneous Min, Max

$A = \{2, 5, 3, 7, 1, 4\}$

Compare elements in pairs:

$$\left.\begin{array}{l} \text{Compare 3 with 7} \Rightarrow 3 < 7 \\ \Rightarrow \text{compare 3 with min and 7 with max} \\ \Rightarrow \min = 2, \max = 7 \end{array}\right\} \text{3 comparisons}$$

$$\left.\begin{array}{l} \text{Compare 1 with 4} \Rightarrow 1 < 4 \\ \Rightarrow \text{Compare 1 with min and 4 with max} \\ \Rightarrow \min = 1, \max = 7 \end{array}\right\} \text{3 comparisons}$$

1 initial comparison + 3(n-2)/2 more comparisons

=1 + 3n/2 -3= 3n/2 - 2 comparisons

Order Statistics

Simultaneous Min, Max

Total number of comparisons:

n is odd: we do $3(n-1)/2$ comparisons

n is even: we do $3n/2 - 2$ comparisons

Thus, in either case the total number of comparisons is at most $3\lfloor n/2 \rfloor$.