

Order Statistics

- Select the i_{th} smallest of n elements (the element with rank i).
 - Minimum: i =
 - Maximum: i =
 - Median: i =
- What is a naive algorithm for this problem?

2

• What is its worst-case running time?

2/25/11 CS380 Algorithm Design and Analysis

Minimum and Maximum

• MINIMUM(A, n)

• How many comparisons are needed?

Max and Min

- How many comparisons are needed to find Max and Min independently?
- Can we do better?

Simultaneous Max and Min

At most 3n/2 comparisons are needed

Analysis

Total number of comparisons when:
 n is odd:

o n is even:

Example

• n = 5, A = <2, 7, 1, 3, 4>

Example

• n = 6, A = <2, 5, 3, 7, 1, 4>

Order Statistics

• RANDOMIZED-SELECT(A, p, r, i)

Example

2/25/11

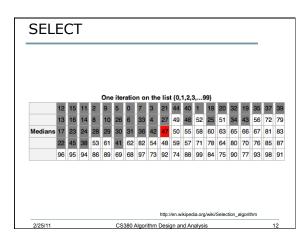
• A: <6, 10, 13, 5, 8, 3, 2, 11>

Selection in worst-case Linear <u>Time</u> • The worst-case for RANDOMIZED-SELECT is n² • Can we do better?

CS380 Algorithm Design and Analysis

10

11





Finding i Largest Numbers

- Problem 9-1: Given a set of *n* numbers, we wish to find the *i* largest in sorted order using a comparison-based algorithm. Find the algorithm that implements each of the following methods with the best asymptotic worst-case running time, and analyze the running times of the algorithms in terms of *n* and *i*.
 - Sort the numbers, and list the *i* largest.
 - Build a max-priority queue from the numbers and call EXTRACT-MAX i times.
 - Use an order-statistic algorithm to find the *i*th largest number, partition around that number, and sort the I largest numbers.

13

2/25/11 CS380 Algorithm Design and Analysis