Medians and Order Statistics

Chapter 9

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?
- What is its worst-case running time?

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:
  - 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

- A: <6, 10, 13, 5, 8, 3, 2, 11>

Selection in worst-case Linear Time

- The worst-case for RANDOMIZED-SELECT is n^2
- Can we do better?

SELECT

One iteration on the list {0,1,2,3,...99}

	12	15	11	2	9	5	0	7	3	21	44	40	1	18	20	32	19	35	37	39
	13	16	14	8	10	26	6	33	4	27	49	46	52	25	51	34	43	56	72	79
Medians	17	23	24	28	29	30	31	36	42	47	50	55	58	60	63	65	66	67	81	83
	22	45	38	53	61	41	62	82	54	48	59	57	71	78	64	80	70	76	85	87
	96	95	94	86	89	69	68	97	73	92	74	88	99	84	75	90	77	93	98	91

http://en.wikipedia.org/wiki/Selection_algorithm

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 ith largest number, partition around that number, and sort the I largest numbers.