CSE421: Design and Analysis of Algorithms	February 6, 2013
Lecture 13-14 Linear Time Median	
Lecturer: Anup Rao	Scribe:

1 Divide and Conquer for Selection

We are given n numbers $x_1, \ldots, x_n$. Our goal is to compute the k'th smallest of the numbers, which we denote by $Sel(k, x_1, \ldots, x_n)$. We could just sort the numbers and find the k'th smallest number that way, but that would take $O(n \log n)$ time. Instead, we shall develop a divide and conquer algorithm.

For any number w, define the sets $S_L(w) = \{i : x_i < w\}, S_E(w) = \{i : x_i = w\}, S_G(w) = \{i : x_i > w\}$. Given such a w, the sets $S_L(w), S_E(w), S_G(w)$ can be computed in time $O(n)$, and further, they allow for a simple recursive algorithm for computing $Sel(k, x_1, \ldots, x_n)$:

1. Compute $S_L(w), S_E(w), S_G(w)$.
2. If $k \leq |S_L(w)|$, output $Sel(k, S_L(w))$, else if $k \leq |S_L(w)| + |S_E(w)|$, output w, else output $Sel(k - |S_L(w)| - |S_E(w)|, S_G(w))$.

One way to find a w that leads to a good recursive algorithm is to pick w uniformly at random from $x_1, \ldots, x_n$. We can show that the probability that $\max\{|S_L(w)|, |S_G(w)|\}$ exceeds $n/2$ is at most $1/2$. This leads to an algorithm whose expected running time satisfies the recurrence $T(n) \leq T(n/2) + cn$, for some constant c, and so $T(n) \leq cn \sum_{k=0}^{\log n} (1/2)^k \leq O(n)$.

We shall try to find a deterministic algorithm. To this end, we compute w as follows. Partition the inputs into sets of size 5. In each such set, we compute the median. We then recursively compute w to be the median of the resulting $n/5$ medians.

Claim 1. $\max\{|S_L(w)|, |S_G(w)|\} < 7n/10$.

Proof Since w is the median of the medians, we have that at least $1/2$ of the medians have value at most w. Thus $n/10$ of the medians have value at most w. For each one of these there are 3 elements in each partition that have value at most w. Thus $|S_L(w)| + |S_E(w)| \geq 3n/10$, and so $|S_G(w)| < 7n/10$. Similar reasoning can be used to bound the size of $S_L(w)$. ■

Thus our overall algorithm satisfies the recurrence:

$$T(n) \leq T(n/5) + T(7n/10) + cn,$$

for some constant c. The k'th level of the recursion tree contributes $cn(1/5 + 7/10)^k = cn(9/10)^k$ to the running time. Thus the overall running time can be bounded

$$T(n) \leq cn \sum_{k=0}^{\infty} (9/10)^k = \frac{cn}{1 - 9/10} = O(n).$$