§1 Lecture 01-16

§1.1 Selection Problem

Given: $x_1, \ldots, x_n \in \mathbb{R}$ (pairwise distinct)

Want: Find the kth smallest.

Sorting leads to $\Theta_(n \log n)$ complexity. But we can do better.

Algorithm of Blum et al.

If $n \leq 5$, sort and exit. Cost ≤ 7. Try it at home. Usually it can be done in 8, but with an extra trick you can do it in 7.

Else

1. Divide all elements into groups of 5 and find the median. Cost $\leq n/5 \times x$ where x is the time to find the median of a group of 5 elements. It isn't obvious but it can be done in 6.
2. Find the median of M, recursively. Cost $\leq T_{n/5}$
3. Compare all items with m, forming sets L, R. Cost $= n - 1$.
4. Case $k \leq |L|$, then find the kth smallest in L. Cost is $T_{|L|} \leq T_{7n/10}$
   Case $k = |L| + 1$, then return m. This has cost of 0.
   Case $k > |L| + 1$, find the $k - |L| - 1$ smallest in R. Cost is $T_{|R|} \leq T_{7n/10}$

$$T_N \leq \begin{cases} 7, & n \leq 5 \\ T_{n/5} + T_{7n/10} + \frac{11}{5}n & n > 5 \end{cases}$$

Now to show inductively that $T_n \leq Cn$.

Proof. Case where $n \leq 5$, then $7 \leq C$, so $C \geq 7$.

Now assume that $T_n Cn$ up to $n - 1$, where $n > 5$.

$$T_{n/5} + T_{7n/10} + \frac{11}{5}n$$

$$\Rightarrow \frac{11}{5} \leq \frac{C}{10} \Rightarrow C \geq 22$$

An improvement can be made by return L and R, so step 3 only has to check $\frac{4}{10}n - 1$ elements.

$$C\frac{9}{10}n + \frac{8}{5}n \leq C_n$$
$$\Rightarrow \frac{8}{5} \leq \frac{C}{10} \Rightarrow C \geq 16$$

□

There we go that is the linear time algorithm.

Tips for guessing the order of an algorithm when preparing to do induction: The subscripts on T summed together are less than 1. So when building the recursion you'll see that the cost at each level decreases by $\frac{1}{10}$. So it's reasonable to assume that it might be linear.

Note 1.1. $T_n =$ maximal cost on any input of size $\leq n$. This implies monotone.

§1.2 Algorithm "FIND" (Hoare)

Very similar to the above algorithm.

Worst case of quick sort. $n + (n-1) + (n-2) + \cdots + (n-(n-1)) = \Theta(n^2)$

$E\{T_n\} \leq \frac{1}{2}(ET_n + E_T 3n/4) + n$.
So $ET_n \leq E_{8n/4} + 2n \leq Cn \Rightarrow \frac{3n}{4} \Rightarrow$

§1.3 Mergesort Induction

$$T_n \leq T_{\lfloor (n/2) \rfloor} + T_{\lceil (n/2) \rceil} + n - 1$$

Claim: $T_n \leq Cn \log_2(n)$

Proof.

$$\begin{cases} T_n \leq T_{\lfloor (n/2) \rfloor} + T_{\lceil (n/2) \rceil} + n - 1 \\ T_0 = 0, T_1 = 0 \end{cases}$$

If $n = 1$, ✓.

If n is even. $T_n \leq 2T_{n/2} + n - 1 \leq 2C(n/2)\log_2(\frac{n}{2}) + n - 1$

$$= cn\log_2 n - cn + n - 1 < n\log_2 n \quad \forall C \geq 1$$

If n is odd.

$$\begin{aligned} T_n &\leq T_{\frac{n+1}{2}} + T_{\frac{n-1}{2}} + n - 1 \\ &\leq C\frac{n+1}{2}\log_2\frac{n+1}{2} + C\frac{n-1}{2}\log_2\frac{n-1}{2} + n - 1 \\ &= C\frac{n}{2}\log_2\frac{n^2-1}{4} + \frac{C}{2}\log_2\frac{n+1}{n-1} + n - 1 \\ &= Cn\log_2 n - Cn + \frac{C}{2} + n - 1 \\ &< n\log_2 n \end{aligned}$$

□

§1.4 Binary Search

Given: Sorted array $x_1, \ldots, x_n$.

Find: x. Return either. 1. $x = x_i$ or 2. $x_i < x < x_{i+1}$

Model: Ternary Oracle. Two inputs, three possible outputs (<, =, >).
Binary Oracle: $x \leq y$ or $x > y$.

BinarySearch(x, i, j)

Cases

1. $i = j$. Ternary Oracle(x, x_i). Exit with either $x = x_i$ or x is not present.
2. $i > j$. This doesn't especially make sense. Exit with "x not present"
3. $i < j$. Let $k = \left\lfloor (\frac{i+j}{2}) \right\rfloor$. Ternary oracle (x, x_k).

$$\begin{cases} \text{Return BinSearch } (x, i, k-1) & x < x_k \\ \text{Return "}x = x_k\text{"} & x = x_k \\ \text{Return BinSearch } (x, k+1, j) & x > x_k \end{cases}$$

$$T_n \leq \begin{cases} T_{n/2} + 1 & \text{n even} \\ T_{(n-1)/2} + 1 & \text{n odd} \end{cases}$$
$$T_1 = 1$$
$$T_0 = 0$$