SOLVING A CERTAIN RECURSION.

MAXIMILIAN JANKE

In the exercise classes we were left with solving the following recursion which described the run-time T of a randomized algorithm for finding the median of an array of n elements: For $n \leq 2$ we have $T(n) \leq Cn$ for some constant C, else we have

$$T(n) \leq \max_{1\leq k\leq n} \frac{1}{n} \sum_{j=1}^{n} \begin{cases} T(j) & \text{for } j \geq k \\ T(n-j) & \text{for } j < k \end{cases} + Cn.$$

To solve the recursion we intended to see via induction that we have $T(n) \leq Kn$ for a constant K yet to be determined. In fact any $K \geq 12C$ will work. [1]

For $n < 2$ it was given that $T(n) \leq Cn \leq Kn$. The aim of this document is to outline how to proceed with the induction step for large $n > 2$.

Inductively we have:

$$\begin{aligned} T(n) &\leq \max_{1\leq k\leq n} \frac{1}{n} \sum_{j=1}^{n} \begin{cases} jK & \text{for } j \geq k \\ (n-j)K & \text{for } j < k \end{cases} + Cn. \\ &\leq \max_{1\leq k\leq n} \frac{1}{n} \left(\sum_{j=1}^{k} (n-j) + \sum_{j=k+1}^{n} j \right) K + Cn \\ &\leq \max_{1\leq k\leq n} \frac{1}{n} \left(k(n-k) + \frac{k(k+1)}{2} + k(n-k) + \frac{(n-k)(n-k+1)}{2} \right) K + Cn \\ &= \max_{1\leq k\leq n} \left(\frac{n+1}{2} + \frac{k(n-k)}{n} \right) K + Cn \end{aligned}$$

One can quickly check that the function $f(k) = \frac{k(n-k)}{n}$ is maximized for $k = \frac{n}{2}$. Indeed this function is concave, we have $\partial_k^2 f = \frac{-2}{n} < 0$, and its first derivative $\partial_k f = \frac{n-2k}{n}$ vanishes for $k = \frac{n}{2}$. We hence have

$$\max_{1\leq k\leq n} f(k) \leq f\left(\frac{n}{2}\right) = \frac{n}{4}.$$

In particular we get:

$$\begin{aligned} T(n) &\leq \left(\frac{n+1}{2} + \frac{n}{4} \right) K + Cn \\ &\leq \left(\frac{11}{12} K + C \right) n \leq Kn. \end{aligned}$$

Date: October 31, 2018.

[1] In fact if we assumed $T(n) \leq Cn$ for all $n \leq n'$ we could choose any $K > 4C$ if n' was large enough. We did not specify the value n' in the exercise.

For the second inequality we use that $n \geq 3$, for the last inequality we need to choose $K \geq 12C$.

As an exercise one may try to analyze the run-time of Quicksort using a similar approach. Of course the argument presented in the lecture notes is shorter and more elegant.