Lecture 7

Swagato Sanyal

Recall that in the lecture we derived the following recurrence for the runtime of the algorithm for finding order statisctics:

$$T(n) \leq an + T(\lfloor n/5 \rfloor) + T(\lfloor 3n/4 \rfloor) \quad \text{for } n \geq 90. \tag{1}$$

We wished to show that there exists a constant c such that for all $n \geq 90$, $T(n) \leq cn$. To do that we chose a c such that

- Base case: $T(90) \leq c \cdot 90$.
- Inductive step: $c \geq 20a$.

We inferred that for such a choice of c, $T(n) \leq cn$ for all $n \geq 100$.

Error: That was an erroneous inference. To see this, consider the inductive step when $n = 91$. Then $\lfloor n/5 \rfloor = 18$ and $\lfloor 3n/4 \rfloor = 68$. In our inductive step deduction we assumed that $T(18) \leq c \cdot 18$ and $T(68) \leq c \cdot 68$. However, we chose c so that only the inequalities $T(90) \leq c \cdot 90$ and $c \geq 20a$ are satisfied. So we cannot make such assumptions.

Fix: One way to resolve this is choosing c such that for each $n' = 1, 2, \ldots, 90$, $T(n') \leq c \cdot n'$, i.e., $c \geq T(n')/n'$. All these 90 constraints, and the constraint $c \geq 20a$ can be satisfied by choosing a large enough c. Notice that for such a choice of c the following is true:

$$T(n) \leq c \cdot n \quad \text{for all } n \geq 1.$$