CS500 Design and Analysis of Algorithm

Average-Case Analysis of QuickSort

Prove that $T(n) = \mathcal{O}(n \log n)$ solves the recurrence

$$T(n) \;\; = \;\; 1/n \cdot \sum\nolimits_{j=1}^{n} T(j) \; + \; T(n-j) \; + \; \mathcal{O}(n) \;\; . \tag{1}$$

First "proof":

$$\begin{aligned} &1/n \cdot \sum\nolimits_{j=1}^{n} \mathcal{O}\big(j \cdot \log j\big) \; + \; \mathcal{O}\big((n-j) \cdot \log(n-j)\big) \; + \; \mathcal{O}(n) \\ &\quad \leq \; \tfrac{2}{n} \cdot n \cdot \mathcal{O}(n \cdot \log n) \; + \; \mathcal{O}(n) \; = \; \mathcal{O}(n \cdot \log n) \;\; . \end{aligned}$$

But then it would similarly follow that $T(n) = \mathcal{O}(n)$ solves Equation (1) as well, which is does not:

$$\begin{aligned} &2/n \cdot \sum\nolimits_{j=1}^{n} \mathcal{O}(j) \; + \; \mathcal{O}(n) \\ &\leq \; 2/n \cdot n \cdot \mathcal{O}(n) \; + \; \mathcal{O}(n) \; = \; \mathcal{O}(n) \;\; . \end{aligned}$$

A correct proof therefore must take care of constants and lower terms otherwise ignored in big-Oh:
Replace $\mathcal{O}(n)$ in Equation (1) with $c \cdot n$; and make the *Ansatz* $T(n) = C \cdot n \log n$.
Next record that, for (w.l.o.g. even) n,

$$\begin{aligned} \sum_{j=1}^{n} j \cdot \log j \;\; &\leq \;\; \sum_{j=1}^{n/2} j \cdot \log(n/2) \; + \; \sum_{j=1}^{n/2} (j + n/2) \cdot \log n \\ &= \;\; n/4 \cdot (n/2+1) \cdot \log(n/\mathbf{2}) \; + \; n/4 \cdot (n/2+1) \cdot \log(n) \; + \; n^2/4 \cdot \log n \\ &= \;\; n^2/2 \cdot \log(n) \; + \; n/4 \cdot \log(n/2) \; \boldsymbol{- n^2/8} \;\; . \end{aligned}$$

Important is not only the constant $\frac{1}{2}$ in front of the asymptotically leading term $n^2 \cdot \log n$, but also the subtracted quadratic term. Because now, indeed, $1/n \cdot \sum_j T(j) + T(n-j) + c \cdot n =$

$$= 2/n \cdot \sum_{j=1}^{n} C \cdot (j \cdot \log j) + c \cdot n \; \leq \; C \cdot n \cdot \log(n) + C/2 \cdot \log(n/2) \underbrace{- C \cdot n/4 \; + c \cdot n}$$

$\leq \; C \cdot n \cdot \log(n)$ for $C > 4c$ and all sufficiently large n.