HASSO PLATTNER INSTITUTE

Heuristic Optimization - Homework 1

Maximilian Brehm & Clemens Frahnow

May 21, 2015

1 PROBLEM (A)

1.1 GIVEN

Algorithm A running on inputs of size n with random variable T_n describing the run time of A on those inputs and

$$E(T_n) \leq 5n^2. \tag{1.1}$$

1.2 ASSUMPTION

We want to show that:

$$P(X \geq n^3) = O(\tfrac{1}{n}). \tag{1.2}$$

1.3 PROOF

Markov's inequality says, if X is any nonnegative, integrable random variable:

$$P(X < 0) = 0 \Rightarrow P(X \geq f(n)) \leq \tfrac{E(X)}{f(n)}. \tag{1.3}$$

The premise $P(T_n < 0) = 0$ is true for T_n since run times can not be negative. Thus, it follows from 1.3:

$$P(T_n \geq n^3) \leq \tfrac{E(X)}{n^3}. \tag{1.4}$$

We know that $E(T_n) \leq 5n^2 \wedge n > 0$ and thus:

$$\tfrac{E(T_n)}{n^3} \leq \tfrac{5n^2}{n^3} = \tfrac{5}{n}. \tag{1.5}$$

From 1.3 and 1.5 we get:

$$P(T_n \geq n^3) \leq \frac{E(T_n)}{n^3} \leq \frac{5}{n} \tag{1.6}$$

$$\Rightarrow P(T_n \geq n^3) \leq \frac{5}{n}. \tag{1.7}$$

We now use the definition of $O(\frac{1}{n})$ to show that $P(T_n \geq n^3)$ is in the class $O(\frac{1}{n})$:

$$f = O(g) \Leftrightarrow \exists c > 0 \exists n_0 > 0 \forall n > n_0 : |f(n)| \leq c|g(n)|. \tag{1.8}$$

Thus, it follows that:

$$\exists c > 0 \exists n_0 > 0 \forall n > n_0 : P(T_n \geq n^3) \leq c \cdot \frac{1}{n} \Leftrightarrow P(T_n \geq n^3) = O(\frac{1}{n}). \tag{1.9}$$

With:

$$\exists c > 0 \exists n_0 > 0 \forall n > n_0 : P(T_n \geq n^3) \leq \frac{5}{n} \leq c \cdot \frac{1}{n}, \tag{1.10}$$

we see that for $c = 5 \wedge n_0 = 1$ $P(T_n \geq n^3) = O(\frac{1}{n})$ is fullfilled and our assumption holds. □

2 Problem (b)

2.1 Given

Algorithm A running on inputs of size n with random variable T_n describing the run time of A on those inputs and

$$E(T_n) \leq 5n^2. \tag{2.1}$$

2.2 Task

Give an example for T_n such that $P(T_n \geq n^3) = \Theta(\frac{1}{n})$.

2.3 Solution

We define Ω with $\Omega = \mathbb{N} \backslash \{0\}$ to be the set of possible run times. We assume n to be at least 1, since there has to be an input for the algorithm. We define P and T_n to be in a way, such that:

$$P(T_n = r) = 0 \text{ for } r \neq 0 \wedge r \neq n^3 \tag{2.2}$$

$$P(T_n = n^3) = \frac{1}{n} \tag{2.3}$$

$$P(T_n = 0) = 1 - \frac{1}{n}. \tag{2.4}$$

Thus (Ω, P) describes a discrete probability space. First, we need to show, that the upper bound for the expected value 2.1 holds:

$$E(T_n) = \sum_{r \in \mathbb{R}} r \cdot P(T_n = r) = n^3 \cdot \frac{1}{n} + 0 \cdot (1 - \frac{1}{n}) + \sum_{r \in \mathbb{R} \wedge r \neq 1 \wedge r \neq n^3} r \cdot 0 \tag{2.5}$$

$$\Rightarrow E(T_n) = n^2 \leq 5n^2 \tag{2.6}$$

$$\Rightarrow E(T_n) \leq 5n^2. \tag{2.7}$$

Now we show that $P(T_n \geq n^3) = \Theta(\frac{1}{n})$. Given our definition, $T_n = n^3$ is the only possible case with a value greater or equal to n^3 and thus:

$$P(T_n \geq n^3) = P(T_n = n^3) = \frac{1}{n}. \tag{2.8}$$

The function $f(n) = \frac{1}{n}$ is obviously in $\Theta(\frac{1}{n})$ and thus

$$P(T_n \geq n^3) = \Theta(\frac{1}{n}). \tag{2.9}$$

3 Problem (c)

3.1 Given

Algorithm A running on inputs of size n with random variable T_n describing the run time of A on those inputs and

$$E(T_n) \leq 5n^2. \tag{3.1}$$

Algoritm A' that executes A until A successfully terminates in $t_0 = 25n^2$ and reruns A if it is not successful in t_0. T'_n that is the run time of A'.

3.2 Assumption

We want to show that

$$P(T'_n \geq n^3) = 2^{-\Omega(n)}. \tag{3.2}$$

3.3 Proof

A' executes A until A returns in at least t_0. For the probability that this happens in a single execution, we can say that:

$$
\begin{aligned}
& P(T_n > 25n^2) && && (3.3)\\
& \leq P(T_n \geq 25n^2) && && (3.4)\\
& \leq \tfrac{E(T'_n)}{25n^2} && \text{Markov's Inequality, as } P(T_n < 0) = 0, n > 0 && (3.5)\\
& \leq \tfrac{5n^2}{25n^2} && \text{Definition of 3.1} && (3.6)\\
& = \tfrac{1}{5}. && && (3.7)
\end{aligned}
$$

We thus know:

$$P(T_n > 25n^2) \leq \frac{1}{5}. \tag{3.8}$$

and

$$P(T_n \leq 25n^2) = 1 - P(T_n > 25n^2) \geq \tfrac{4}{5}. \tag{3.9}$$

We can thus express the number of executions of A in A' with a geometrically distributed random variable X with $p \geq \frac{4}{5}$. For the number of times k that A' needs to execute A at least to have a run time of at least n^3, we know that:

$$k \cdot 25n^2 \geq n^3 \tag{3.10}$$

$$k \geq \tfrac{n}{25}. \tag{3.11}$$

For the smallest value $k = \lceil \frac{n}{25} \rceil$, the run time may be smaller than n^3 as the k.th run may be successful with a total run time smaller than n^3 (k is the smallest number of executions of A that can exceed a run time of n^3). Even so, we have a close upper bound to estimate $P(T'_n \geq n^3)$:

$$P(T'_n \geq n^3) \tag{3.12}$$

$$\leq P(X \geq k-1) \quad \text{Definition of k} \tag{3.13}$$

$$\leq P(X \geq \tfrac{n}{25} - 1) \quad \text{As in 3.11} \tag{3.14}$$

$$= (1-p)^{\frac{n}{25}-1} \quad \text{We run } k-1 \text{ times unsucessfully, and do not care about the rest} \tag{3.15}$$

$$\leq \tfrac{1}{5}^{\frac{n}{25}-1}. \quad \text{As in 3.8} \tag{3.16}$$

Now we see that:

$$P(T'_n \geq n^3) \leq \tfrac{1}{5}^{\frac{n}{25}-1} = 2^{-log_2(5)(\frac{n}{25}-1)} = 2^{-\frac{log_2(5)}{25}n - log_2(5)} = 2^{-\Omega(n)} \tag{3.17}$$

as $\frac{log_2(5)}{25}n - log_2(5) = \Omega(n)$. □