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

$$E(T_n) \leq 5n^2.$$  \hfill (1.1)

1.2 ASSUMPTION

We want to show that:

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

1.3 PROOF

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

$$P(X < 0) = 0 \Rightarrow P(X \geq f(n)) \leq \frac{E(X)}{f(n)}.$$  \hfill (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:

$$P(T_n \geq n^3) \leq \frac{E(T_n)}{n^3}.$$  \hfill (1.4)

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

$$\frac{E(T_n)}{n^2} \leq 5n \wedge \frac{5}{n} = \frac{5}{n}.$$  \hfill (1.5)
From 1.3 and 1.5 we get:

\[
P(T_n \geq n^3) \leq \frac{E(T_n)}{n^3} \leq \frac{5}{n}.
\]  
(1.6)

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

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

\[
f = O(g) \leftrightarrow \exists c > 0 \exists n_0 > 0 \forall n > n_0 : |f(n)| \leq c|g(n)|.
\]  
(1.8)

Thus, it follows that:

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

With:

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

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

\[\Box\]

## 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

\[
E(T_n) \leq 5n^2.
\]  
(2.1)

### 2.2 TASK

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

### 2.3 SOLUTION

We define \(\Omega\) with \(\Omega = \mathbb{N}\setminus\{0\}\) 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:

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

\[
P(T_n = n^3) = \frac{1}{n}
\]  
(2.3)

\[
P(T_n = 0) = 1 - \frac{1}{n}.
\]  
(2.4)

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

\[
E(T_n) = \sum_{r \in \mathbb{R}} r \cdot P(T_n = r) = n^3 \cdot \frac{1}{n} + 0 \cdot (1 - \frac{1}{n}) + \sum_{r \in \mathbb{N} \setminus \{0\} \setminus \{n^3\}} r \cdot 0
\]  
(2.5)

\[
\Rightarrow E(T_n) = n^2 \leq 5n^2
\]  
(2.6)

\[
\Rightarrow E(T_n) \leq 5n^2.
\]  
(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:

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

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

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

$$E(T_n) \leq 5n^2.$$  \hspace{1cm} (3.1)

Algorithm $A'$ 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

$$P(T'_n \geq n^3) = 2^{-\Omega(n)}.$$  \hspace{1cm} (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:

$$P(T_n > 25n^2) \leq P(T_n \geq 25n^2) \leq \frac{E(T'_n)}{25n^2}$$

Markov’s Inequality, as $P(T_n < 0) = 0$, $n > 0$  \hspace{1cm} (3.5)

$$\leq \frac{5n^2}{25n^2}$$

Definition of 3.1  \hspace{1cm} (3.6)

$$= \frac{1}{5}.$$  \hspace{1cm} (3.7)

We thus know:

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

and
\[
P(T_n \leq 25n^2) = 1 - P(T_n > 25n^2) \geq \frac{4}{5}. \tag{3.9}
\]

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:

\[
k \cdot 25n^2 \geq n^3 \quad \Rightarrow \quad k \geq \frac{n}{25}.
\tag{3.10}
\]

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 \(P(T'_n \geq n^3)\):

\[
P(T'_n \geq n^3) \leq P(X \geq k - 1) \quad \text{Definition of } k \tag{3.12}
\]
\[
\leq P(X \geq \frac{n}{25} - 1) \quad \text{As in 3.11} \tag{3.13}
\]
\[
= (1 - p)^{\frac{n}{25} - 1} \quad \text{We run } k - 1 \text{ times unsuccessfully, and do not care about the rest} \tag{3.14}
\]
\[
\leq \left( \frac{4}{5} \right)^{\frac{n}{25} - 1} \quad \text{As in 3.8} \tag{3.15}
\]

Now we see that:

\[
P(T'_n \geq n^3) \leq \left( \frac{4}{5} \right)^{\frac{n}{25} - 1} = 2^{-log_2(5)\left( \frac{n}{25} - 1 \right)} = 2^{-\frac{log_2(5)}{25} n - log_2(5)} = 2^{-\Omega(n)} \tag{3.16}
\]

as \(\frac{log_2(5)}{25} n - log_2(5) = \Omega(n)\). \hfill \square