# CSC263: Problem Set 1

Angus Fung,[1] Syed Kamran,[1] Michelle Leung[1]

[1]Division of Engineering Science, University of Toronto,

## Question 1

(a) Let the number of list accesses (excluding return statements) be the basic operation under consideration for the analysis. Since the for loop runs at most $n$ times, then a pessimistic upper bound counting said basic operation is at most $n$ for any input of a fixed size $n$. Furthermore, the worst-case running time would be where $x \notin A$ - the value that is being searched is not in the list. Since this is the absolute worst case, all other cases must have run times $T(n) \leq n$. As such, the algorithm will access every element of the list, and $T(n) = n = \mathcal{O}(n)$.

(b) Let there be $n$ number of elements in list $A$. The *search* algorithm must terminate after $n$ iterations - that is, after accessing every element in $A$. If on the $n$th iteration - so the last element of the list - it succeeds, then the worst-case is exactly $n$ and $T(n) = n = \Omega(n) = \mathcal{O}(n) = \Theta(n)$. Although, even if it does not succeed (i.e (a)) and $x \notin A$, the worst-case is still exactly $n$ and the aforementioned statements remain valid.

(c) If we measure running time by the *number of items visited* by the for loop, then the probability of $x$ being in position $t$ is the probability that $x$ is not in the previous positions and the probability that $x$ is in the current position: $\frac{1}{263} \cdot (\frac{262}{263})^{t-1}$. The only caveat being the last element in the list ($T = n$) - the probability is simply the probability that $x$ is not in the first $n-1$ elements. Therefore, the actual value of the $n$th element need not matter as the $n$th element will be visited simply on the basis that none of the previous elements are $x$. Because of this, we do not need to include the probability of $x \notin A$. If we let $T$ be the random variable, then the average case running time can be calculated using the expected value:

$$
\begin{aligned}
\mathbb{E}[T] &= \sum_{t=1}^{n} t \cdot P[T = t] \\
&= 1 \cdot \frac{1}{263} + 2 \cdot \frac{262}{263} \cdot \frac{1}{263} + ... + n \cdot \left(\frac{262}{263}\right)^{n-1} \\
&= \sum_{t=1}^{n-1} t \cdot \left(\frac{262}{263}\right)^{t-1} \frac{1}{263} + n \cdot \left(\frac{262}{263}\right)^{n-1}
\end{aligned}
$$

If we let $r = \frac{262}{263}$, and using the fact that

$$
\sum_{t=1}^{n-1} tr^{t-1}(1-r) = \frac{1}{1-r} - r^n(n + \frac{1}{1-r}) - nr^{n-1}(1-r)
$$

we obtain:

$$
\begin{aligned}
\mathbb{E}[T] &= 263 - \left(\frac{262}{263}\right)^{n} (n + 263) - n\left(\frac{262}{263}\right)^{n-1} \frac{1}{263} + n\left(\frac{262}{263}\right)^{n-1} \\
&= 263 - \left(\frac{262}{263}\right)^{n} (n + 262)
\end{aligned}
$$

In fact, in the limit of large $n$, the exponential term vanishes and $\mathbb{E}[T] = 263 = \Theta(263) = \Theta(1)$.

(d) Here, $x$ is chosen uniformly at random from a larger range, 1 and 363. The average-case analysis for this new distribution is the same as that of $c$), except with a new value of $r$, which is $\frac{1}{363}$. Therefore, substituting $r = \frac{372}{373}$ into the result in (c)

$$
\mathbb{E}[T] = 373 - \left(\frac{372}{373}\right)^{n} (n + 372)
$$

# Question 2

(a) Given that the list is of length $n$ where $n$ is a perfect square, and each element is chosen uniformly and independently at random from a set $\{1, 2, ..., \sqrt{n}\}$ hence for each element, the probability of choosing the maximum $\sqrt{n}$ from the set is $\frac{1}{\sqrt{n}}$. The probability that the algorithm returns the correct maximum is equal to the total probability minus the probability that the maximum $\sqrt{n}$ does not exist in the list of $n$ elements.

The probability that $\sqrt{n}$ does not exist in the list, $q$, is equal to the number of non $\sqrt{n}$ choices (so, $\sqrt{n} - 1$) over the total possible choices ($\sqrt{n}$), chosen independently for each element (with a total of $n$ elements). Hence, $q = (\frac{\sqrt{n}-1}{\sqrt{n}})^n$. Therefore, the probability that *cheater* returns the right maximum value of the list, $p$ is:

$$p = 1 - q$$

$$p = 1 - (\frac{\sqrt{n}-1}{\sqrt{n}})^n$$

(b) Using the same input distribution in (a), we first calculate the probability $P_k$ for all $k$ as follows:

$$P_1 = (\frac{1}{\sqrt{n}})(\frac{1}{\sqrt{n}})^{n-1}$$

where $(\frac{1}{\sqrt{n}})$ is the probability of getting 1 for first position, and $(\frac{1}{\sqrt{n}})^{n-1}$ is the probability that all the other elements are less than or equal to 1.

Generalizing, we can say that $P_k$ is equal to:

$$P_k = (\frac{1}{\sqrt{n}})(\frac{k}{\sqrt{n}})^{n-1}$$

Summing all the probabilities from $k = 1$ to $k = \sqrt{n}$ we have

$$\begin{aligned}
\sum_{k=1}^{\sqrt{n}} P_k &= \sum_{k=1}^{\sqrt{n}} (\frac{1}{\sqrt{n}})(\frac{k}{\sqrt{n}})^{n-1} \\
&= (\frac{1}{\sqrt{n}})(\frac{1}{\sqrt{n}})^{n-1} \sum_{k=1}^{\sqrt{n}} k^{n-1} \\
&= (\frac{1}{\sqrt{n}})^{n} \sum_{k=1}^{\sqrt{n}} k^{n-1} \\
&= \Theta((\frac{1}{\sqrt{n}})^{n})\Theta(\sqrt{n}^{n}) \\
&= \Theta(1)
\end{aligned}$$