# CSC263H Tutorial 1

Sample Solutions

Winter 2024

Consider the following algorithm to find the maximum element in a list $L$ of integers (indexes starting from 1).

```
def FIND-MAX(L):
        max = -oo  # minus infinity
        for k = L.length,...,1:
            if L[k] > max:
                 max = L[k]
        return max
```

For this problem, we are interested in the number of times that variable $max$ gets assigned a value.

1. Give a tight-bound on the worst-case number of times that $max$ is assigned a value by algorithm `FIND-MAX`. Show your work.

   **Solution:** Let $t_n$ denote the number of times that $max$ is assigned a value by algorithm `FIND-MAX`, where $n$ is the length of $L$.
   The worst-case happens when $L[k] > max$ is true for all indexes $k$ of $L$.

   $L[k] > max$ is true at most once for each element, and there are $n$ elements in $L$. This implies that for all inputs of size $n$, $t_n$ is less than or equal to $n$. So $t_n \in \mathcal{O}(n)$

   Let $L = [n, n-1, ..., 1]$. Then $L[k] > max$ is true for every element. This implies that for this input, $t_n$ is greater than $n$. So $t_n \in \Omega(n)$.

   So $t_n \in \Theta(n)$.

2. Try to give a tight-bound on the average-case number of times that $max$ is assigned a value by algorithm `FIND-MAX`? Show your work – in particular, you should try to complete the following steps:

   (a) Define the space of all inputs of size $n$.
   **Solution:** $S_n = \{L : L \text{ is a list of integers of size } n\}$.

   (b) Identify an appropriate probability distribution over inputs and state it clearly.

   This is what we should do in general when we are trying to identify a probability distribution for algorithm inputs:
   Think about inputs and whether any of them represent special cases. We may not know how likely each one is, in which we you can parametrize the answer, assign unknown but fixed probabilities to each special case, and then make the rest of the **regular** cases uniform. As it can be inferred

from the above process, if there are no special cases, we can use uniform distribution.

NOTE: In this course we do not expect you to be able to identify probability distributions for complex inputs (we provide the distribution in such cases). But for simple cases, such as this one, you should be able to identify an appropriate distribution.

**Solution:** There is no special case for the inputs, so we can assume that permutations of integers can occur in the list equally likely.

(c) Define the random variable $M$.
Hint: Define the random variable for quantity of interest (that is, the quantity for which we want to calculate the expected value).

**Solution:** $M$: number of times the statement $max = L[k]$ is executed on input $L$.

(d) Figure out possible values of $M$ and try to calculate the probability of the occurrence of each value.

NOTE: You may notice that the probability of each value is not obvious to compute.
This does not mean it's impossible; you may be able to work on it until coming up with a reasonable expression.
But I'm going to drop it here and use another technique (see Question 3).

**Solution:** Possible values for $M$ are $1, 2, 3, ..., n$.
However, computing the the probability of each value is not obvious. So we will move on to question 3.

(e) If you were able to complete part (d), plug-in the results you have came up with into expected value formula, simplify your final answer, and provide a tight-bound for it.
If not, move on to the next question.

3. Try to come up with indicator random variables for $M$ (from part (d) of question 2).
Then use the indicator random variables to compute the expected value of $M$.

**Solution:** Let

$$M_i = \begin{cases} 1 & \text{if } L[i] \text{ is greater than } L[i+1] \text{ and } L[i+2] \text{ and } L[i+3] \text{ and ... and } L[n] \\ 0 & \text{otherwise.} \end{cases}$$

Using this indicator random variables, we have that $M = M_1 + ... + M_n$ (the number of times that $max = L[k]$ is executed is equal to the number of times that $L[k]$ is larger then each of the following elements in $L$). Then, as explained above:

$$\mathbb{E}[M] = Pr[M_1 = 1] + ... + Pr[M_n = 1]$$

But what's $Pr[M_i = 1]$? It's the probability that $L[i]$ is greater than every following element. Since the distribution over $L$ uniform, this is simply equal to $1/(\text{number of elements in } L[i...n])$, which is equal to $1/(n - i + 1)$ (There are $n - i + 1$ positions from index $i$ to index $n$, and the greatest value among the values stored at these positions is equally likely to be in any one of these positions. That is, the probability that the greatest value be at an arbitrary index $k$, $i \leq k \leq n$ is $1/(n - i + 1)$).
(Quick sanity check: when $i = n$, the probability is $1/(n - n + 1) = 1/1$, which is correct: $max = L[n]$

is always executed for every input. When $i = 1$, probability is $1/(n-1+1) = 1/n$, which is also correct: $max = L[0]$ is executed only if $L[0]$ is the largest element, which happens with probability $1/n$.)

So now we have:

$$\mathbb{E}[M] = \sum_{i=1}^{n} Pr[M_i = 1] = \sum_{i=1}^{n} 1/(n-i+1) = \sum_{j=1}^{n} 1/j = H_n \in \Theta(\log n)$$

(Because the $n$-th Harmonic number $H_n$ converges to $\log n$ as $n$ tends to infinity.)