# Problem Set : Week 4

Due: 2pm on 3/9/25

## Instructions

- Discussion is allowed and infact encouraged
- Answers must be written by yourself.
- All sources that are used to reach the solution must be mentioned.

---

(1) Show that in any sub-tree of a min-heap, the root of the subtree contains the smallest value occuring anywhere in that subtree. [3]

(2) Where in a min-heap can the largest element reside? Assume that all the elements are distinct. [2]

(3) Recall the pseudocode for Build-Min-Heap:

```
Build Min-Heap (A)
1.  A.heap-size = A.length
```

2. for $i = \lfloor A.length/2 \rfloor$ down to 1

3. Min-Heapify (A).

Why do we want the loop index $i$ (in line 2) to decrease from $\lfloor A.length/2 \rfloor$ to 1 rather than increase from 1 to $\lfloor A.length/2 \rfloor$? [2]

(4) Recall the Activity Selector Problem:

Input: $S = [s_1, \ldots, s_n]$, $T = [t_1, \ldots, t_n]$

Intuitively, $s_i$ and $t_i$ are the start and end times of activity $i$.

Where $T$ is in increasing order

Output: $A_0 \subseteq [n]$ s.t.

$$|A_0| = \max_{A \subseteq [n]} \left\{ |A| : A = \{i_1, \ldots, i_{|A|}\} \text{ s.t. } s_{i_j} \geq t_{i_{j-1}} \ \forall j \geq 2 \right\}$$

One Greedy Algorithm to solve this would be the following:

Greedy-Activity-Selector $(s, f)$

1. $n = s.length$
2. $A_0 = \{1\}$

```
3.  k = 1
4.  for m = 2 to n
5.      if s_m ≥ f_k
6.          A_0 ← A_0 ∪ {m}
7.          k = m
8.  return A_0
```

Instead, suppose we always select the last activity to start that is compatible with all the previously selected activities. Write a pseudocode for it and argue correctness. [5]

(5) If a, b are the codewords of max length in a Huffman code, why should the length of a and b be same? [1]

(6) The fibonacci sequence is described as follows: $f_0 = 1$, $f_1 = 1$

and $\forall\ n \geq 2 \quad f_n = f_{n-1} + f_{n-2}$.

What is the optimal code for $\Sigma = \{a_0, \ldots, a_n\}$

if $\forall i \in \{0, 1, \ldots, n\}$, $a_i \cdot freq = f_i$ ? [3]

(7) Prove that if we arrange the characters in an alphabet in monotonically decreasing order of frequencies, then there is an optimal code whose codeword lengths are monotonically increasing. [4]

—