1 Recursion [20 marks]

(A) Solve the following recurrence relation. Show all working. You can leave the n-th harmonic function as H_n in your answer. Show all working. **[10 marks]**

$$\begin{cases} T(n) = \frac{1}{n}(T(0) + T(1) + \ldots + T(n-1)) + 5n, \\ T(0) = 0. \end{cases}$$

(B) Solve the following recurrence relation. You can assume that n is a power of 7. Show all working. **[10 marks]**

$$\begin{cases} T(n) = T(\left\lfloor \frac{n}{7} \right\rfloor) + \log_3(n), \\ T(1) = 0. \end{cases}$$

2 Algorithm Analysis [20 marks]

You are presented with a brute force string-search algorithm given below.

Input: text t of length n and word p of length 3.

Output: a position at which we have p in the text. If p can be found in the text several times then we take the first occurrence.

Algorithm:

i) We start from the beginning of the text index $i = 0$

ii) If $i > n - 3$ then terminate and return FALSE.

iii) Query if the sub-string $t[i, ..., i + 2]$ is the same as p.

v) If yes, return i and terminate. Otherwise set i to $i + 1$ and go to step ii)

Note that we are looking only for the first match.
The pseudocode is given below:

Algorithm 1 String-search algorithm

```
function FIND(string t[0..n − 1], word p of length 3)
    i ← 0
    for i ← 0 to n − 3 do
        if [t[i, ..., i + 2]] = p then
            return i
    return FALSE
```

(A) Find the asymptotic running time in the best case. [**5 marks**]

(B) Find the asymptotic running time in the worst case.[**5 marks**]

(C) Find the asymptotic running time in the average case. You may assume that there is at most one occurrence of p in the text, and any such occurrence (or lack thereof) is equally likely. [**10 marks**]

Justify your answers and give the best (ie closest) possible asymptotic bound.

3 Binary Search Trees [20 marks]

(A) Perform the linear time algorithm to construct a maximum heap on the following array. Please show the heap structure after each iteration. [**5 marks**]

$$A = [0, 54, 93, 2, 12, 32, 40, 16, 25, 51].$$

(B) Consider the same array A above. Is A a minimum heap? Justify your answer by briefly explaining the min-heap property. If A is not a min-heap, then restore the min-heap property and show the restored array A'. [**5 marks**]

(C) Given the array A' with min-heap property, delete the min value of A'. Please describe the heapification process and the outcome of the deletion operations on the min-heap constructed using A'. [**5 marks**]

(D) Prove that the height of a complete binary tree with n nodes is exactly $\lceil \lg(n+1) \rceil - 1$ using mathematical induction. [**5 marks**]

4 Programming Question [20 marks]

Question. There are n different courses online. Each course c_i comes with a duration d_i, and last day constraint l_i, denoted as [c_i,d_i,l_i] for all $0 \leq i \leq n$. This indicates that the course c_i should be taken continuously for d_i days and must be finished before or on the day l_i. Please write a program to **find the valid longest path, and among paths with equal length find the shortest total duration**. A learning path can start and end at any course on the list. Assume that we cannot take two or more courses simultaneously.
Input. The first line of the input is a single integer m, denoting the number of examples to follow. Each of the next m lines is a separate example. For each example, the first element is the total number of different courses n; and the second element is a sequence of arrays. Each array [c_i,d_i,l_i] consists of three elements, a course ID, a duration, and a last day constraint for all $0 \leq i \leq n$.
Output. There should be m lines in the output file, denoting the answer for each example scenario. For each example, the first element is the total number of courses in the developed learning path p; and the second element is the total duration of the path p.

- **Example Input:**

```
1
4,[['A',150,200],['B',200,1400],['C',1000,1200],['D',2000,3100]]
```

- **Example Output:**

```
3,1350
```

- **Explanation:** We take course A, which finishes on the 150-th day. We then take course C, which ends at the 1150-th day. Lastly, we take course B, which ends on the 1350-th day. Taking course D adds up to the 3350-th day, which violates its last day constraint of 3100. Thus, [['A', 150, 200], ['C', 1000, 1200], ['B', 200, 1400]] is a valid learning path of 3 courses, with a total duration of 1350 days.

Hint.

- You may consider **sorting the courses** based on their **last day constraint**.
- You may consider developing the path by adding courses one at a time with **sorted courses** based on their **duration** by using a **binary heap**.
- You may keep track of the **number of courses** and the **total duration** up to the current path.

Notes.

- You can use Java, C,C++, PyPy, Python 3, Go, Rust, F#, Javascript.
- You need to implement your own binary heap.
- There is a limit of 15 submission attempts for this question. This is to prevent spamming the automarker. You may test your program carefully before submitting it to the automarker.
- Small sample input and its corresponding output are provided on Canvas. Nonetheless, you should test your program with as many different inputs as possible. Feel free to share your developed test cases on Piazza.
- The last submission before the assignment deadline will be the one marked.
- Please do not share your code with other students.