**CSE102: Data Structures and Algorithm**
**Assignment-3**
**Max Marks**: 60

---

**Instructions**

- Use C++ programming language only.
- Plagiarism might be checked.
- Upload a Zip folder named <RollNo>_<YourName>. If not named appropriately, it will not be evaluated.
- For the coding questions, you must ensure that your code compiles and runs otherwise these will be awarded **zero marks**.
- Students might be required to give a demo of their codes.
- Inside the Zip folder, there should be files named, `<RollNo>_Q2.cpp`, `<RollNo>_Q3.cpp`, `<RollNo>_Q4.cpp`, `<RollNo>_<YourName>.pdf`.
- `<RollNo>_Q2.cpp` - Code for Q2 (Exclude object files)
- `<RollNo>_Q3.cpp` - Code for Q3 (Exclude object files)
- `<RollNo>_Q4.cpp` - Code for Q4 (Exclude object files)
- `<RollNo>_<YourName>.pdf` - Report including answers for Q1, Q2 ,Q3, Q4
- For coding questions, please provide a brief algorithm description along with an analysis of the time and space complexity.

---

1. (20 points) For each statement, indicate whether it is possible to uniquely identify a tree for the given traversal combinations and provide a proper reason for the answer(Marks will be awarded if the correct reason is mentioned).

   (a) Postorder and Level-order

   (b) Inorder and Preorder

   (c) Inorder and Level-order

   (d) Inorder and Postorder

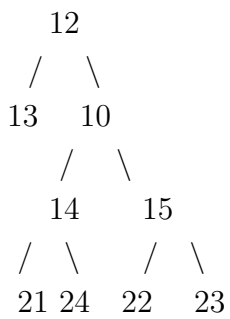
   (e) Postorder and Preorder

   For the cases where traversal combinations can uniquely identify a tree, provide pseudocode to construct a tree from those traversal combinations as input.

2. (15 points) In a sprawling urban landscape meticulously organized in a hierarchical structure of junctions and pathways, with each junction branching off into two distinct pathways, Navnoor, a notorious serial bomber, has detonated a bomb at a crucial junction. Pranav, the police detective, aims to stop Navnoor by mapping out the sequence of junctions affected by the blast. Return the sequence of junctions in the order in which the explosion will impact them.

   The explosion will propagate steadily to connected junctions only. Each junction is impacted by the blast in equal time intervals. A junction is affected by the explosion only once.

   Input :




   Target Node : 14

   Input: 12 13 10 N N 14 15 N N N N 21 24 22 23 (Level order input)

   Output :

   14

   $21, 24, 10$

   $15, 12$

   $22, 23, 13$

   First, the initial bomb explodes at node 14, affecting its immediate neighbours (21, 24, 10). Following this initial detonation, the impact spreads to adjacent junctions sequentially. This process continues until the explosion has impacted all junctions.

3. (10 points) There is a binary tree given to you. Each of the nodes represents a student in the class. You are playing a game where you need to team up with one of the students in class. The team can be formed in such a way that in a team, both students can't have the same parent in the tree and should be at the same level in the binary tree.

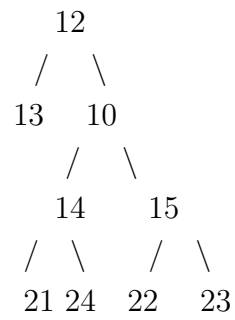
   **Input:** We have given you some pairs of students. You need to find out whether the team is valid or not.

   **Output:** 'Yes' if the team is valid or 'No' if the team is invalid.

(a) Optimal solution of $O(n)$ for checking the team validity.

(b) How many tree traversals are required to solve the above problem?

Example Input :




Target Node : 14

Input: 12 13 10 N N 14 15 N N N N 21 24 22 23 (Level order input)

Query 1: 24 and 22

Output 1: Yes

Query 2: 14 and 15

Output 2: No

For query 1: 24 and 22 are at the same level but don't have the same parent.

For query 2: 14 and 15 both have the same parent.

4. (15 points) Jujutsu High has 3 freshmen, Yuji Itadori, Magumi Fusigoro, and Nobara Kugisaki. They are assigned a list of tasks they must finish together. Each task has a unique ID. However, each has a weird way of choosing their task from the remaining tasks in the list.

   1) Yuji Itadori: Always chooses the tasks with the max or min ID. Min or Max is decided randomly by him (assume 50% probability for each min or max).
   2) Megumi Fusigoro: He will choose a 'k' and call the kth smallest ID (Kth ID when all IDs are arranged in ascending order). K is chosen randomly from the total tasks available in the list ($1 \leq K \leq$ Total tasks).
   3) Nobara Kugisaki: He will always choose the median ID. If there are 'n' tasks in the list, the median will be defined as $(n+1)/2$ (take floor value)th ID when all of them are arranged in ascending order. (For both odd and even 'n' )

Sataru Gojo wants to digitize this process of choosing tasks and is asking for your suggestions on the data structure that would be ideal for this scenario. Your task is to design a data structure that supports the query of each of the three freshmen. The data structure should be highly efficient. It should return the task ID in $O(\log(n))$ for all 3 scenarios.

**Note:-** Once the task is assigned to someone, it can't be reassigned to someone else. It must be deleted from your data structure. Taking input elements, setting them in your data structure can take O(n) or $O(n \log(n))$, but query execution should be in $O(\log(n))$.

**Input example:** a list of unique task IDs = [34, 56, 12, 44, 10] and query 1, 2, or 3. Return the task ID for the corresponding candidate.

Input Query: 1
Output: 56 (Max element, 56 is no longer available in the list after query 1)

Input Query: 2
Enter k: 3
Output: 34 (3rd smallest in the list.)

Input Query: 3
Output: 12 (Initial length was n= 5; after removing 2 elements, the new length is n=3. median = (3+1)/2 = 2, So the 2nd smallest element is 12.)