Course: ESO207A – Data Structures and Algorithms

Indian Institute of Technology Kanpur

Programming Assignment 3

March 20, 2025

Important guidelines for submission

- It is only through the assignments that one learns the most about the algorithms and data structures. You are advised to refrain from searching for a solution on the Internet or in a notebook.
- Refrain from collaborating with the students of other groups or your friends. If any evidence is found that confirms copying, the penalty will be very harsh. Refer to the website at the link: https://cse.iitk.ac.in/pages/AntiCheatingPolicy.html regarding the departmental policy on cheating.
- **The assignment is to be submitted on Gradescope. You must submit one C file named as *assign3.c* for each question. This is EXTREMELY IMPORTANT. The code is checked with an autograder, and a different file name will fail to pass the test cases.**
- **DEADLINE** for this assignment submission is 24th March 23:00.
- In case of any issue related to this assignment, please contact Eswar (esraghava21@iitk.ac.in) and Krutuparna (krutuparna21@iitk.ac.in)
- **Note**: Write the output functions for tasks as function1(), function2(), function3(), function4() respectively. Refer the sample code for more details.

Tasks to be done

1 Random BST

1.1 Part A

1.1.1 Problem statement

You will be simulating a Binary Search Tree (BST) in this task. You will be given an array num of n integers and you have to insert them sequentially to the BST. Initially the BST is empty and you will be adding the integers as leaf nodes. During insertion, you must not modify the existing structure of the tree.
Your task is to find the height of the resulting tree after all insertions.
The expected time complexity for this task is $O(n^2)$ where n is the size of array

1.1.2 Input Format

- The first line contains an integer n - the size of the array num.
- The second line contains n integers (space separated) - The elements of the array num

1.1.3 Output Format

Output the height of the BST.

1.1.4 Constraints

- $1 \leq n \leq 10^4$
- $-10^9 \leq num[i] \leq 10^9 \ \forall i$

1.1.5 Example

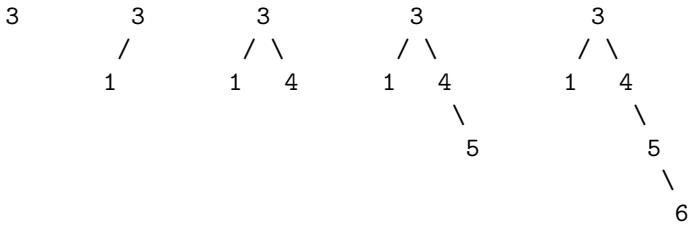
Input

```
5
3 1 4 5 6
```

Output

```
4
```

Explanation
The BST will updated as,



The height of the BST is 4

2 2-3 Tree

2.1 Problem Statement

You will be simulating a 2-3 tree in this task. A node of a 2-3 tree can be described as,

p_v	p_1	k_1	p_2	k_2	p_3

Figure 1: A 2-3 Tree Node

p_v: Index of the parent node if present, or -1 if it is the root node.

p_1: Index of left child node if present, or -1 if absent. This child node has keys smaller than k_1.

k_1: First key value.

p_2: Index of middle child node if present, or -1 if absent. This child node has keys between k_1 and k_2.

k_2: Second key value, or -1 if only one key is present.

p_3: Index of right child node if present, or -1 if absent. This child node has keys larger than k_2.

You will be constructing a 2-3 tree with the given array of nodes and will be implementing two features of a 2-3 tree:

1. search(x) - which returns 1 if x is present or 0 otherwise.
2. successor(x) - which returns the least key that is greater than x where x is present in the tree or 0 if no key is greater than x.

The queries will be of type q_i x, where:

- $q_i = 1$ represents a search query
- $q_i = 2$ represents a successor query

You have to output all the query results in separate lines.
The expected time complexity for this task is $O(n + qlogn)$ where n is the number of nodes and q is the number of queries.
Note: You have to implement the algorithms of 2-3 tree taught in the class. Your code will also be checked manually for this task.

2.2 Input Format

- The first line contains two integers n and q - the number of nodes in the 2-3 tree and the number of queries, separated by space.
- The next n lines contain nodes of the 2-3 tree. Each line is specified by 6 integers (p_{vi}, p_{1i}, k_{1i}, p_{2i}, k_{2i}, p_{3i}), separated by spaces, as discussed above. (The node at index 0 will be the root node always)
- The next q lines contain queries. Each query consists of two integers q_i and x, as discussed above, separated by space.

2.3 Output Format

Output q lines, each line containing a single integer - the answer of the corresponding query.

2.4 Constraints

- $1 \leq n \leq 10^5$
- $1 \leq q \leq 10^5$
- $1 \leq p_1, p_2, p_3, p_v < n$ or -1 for all nodes in the tree
- $1 \leq k_1, k_2 \leq 10^9$ or -1 for all nodes in the tree
- For each query, $q_i = 1$ or 2 and $1 \leq x \leq 10^9$

2.5 Example

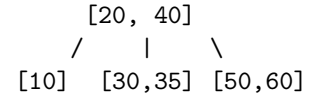
Input

```
4 4
-1 1 20 2 40 3
0 -1 10 -1 -1 -1
0 -1 30 -1 35 -1
0 -1 50 -1 60 -1
1 10
1 15
2 35
2 60
```

Output

```
1
0
40
0
```

Explanation
Given 2-3 tree can be represented as,



The query results are self explanatory

3 City guide

3.1 Problem Statement

You are given a city represented as a connected graph. The city has one office node, one home node, and m market nodes. All other nodes are regular city locations. Each edge in the graph represents a road of length 1. Your task is to find the minimum distance route from the office to a market and then to home, considering that all markets sell the same items. You need to determine the minimum total distance (office to market to home) and identify all markets that result in this minimum distance.
The expected time complexity for this task is $O(n + e)$ where n is the number of nodes and e is the number of edges.

3.2 Input Format

- The first line contains three integers n, e, and m - the number of nodes, edges, and markets respectively.
- The next e lines contain two integers u and v each, representing an edge between nodes u and v.
- The next line contains two integers o and h - the office and home nodes respectively.
- The last line contains m integers - the nodes marked as markets.

3.3 Output Format

Output two lines:

- The first line should contain two integers - the minimum total distance and the number of markets that result in this minimum distance.
- The second line should contain space-separated integers - all market nodes that result in the minimum distance, in ascending order.

3.4 Constraints

- $2 \leq n \leq 10^5$
- $1 \leq e \leq \min(n(n-1)/2, 2 \times 10^5)$
- $1 \leq m \leq n - 2$
- $0 \leq u, v, o, h \leq n - 1$

3.5 Example

Input

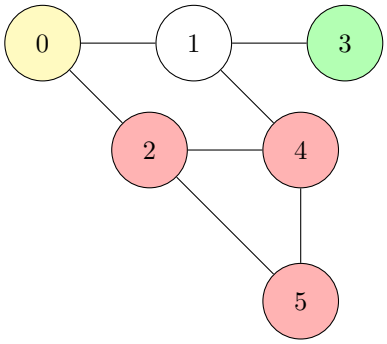
```
6 7 3
0 1
0 2
1 3
1 4
2 4
2 5
4 5
0 3
2 4 5
```

Output

```
4 2
2 4
```

Explanation

The city can be represented as,



In this example:

- Node 0 (yellow) is the office.
- Node 3 (green) is the home.
- Nodes 2, 4, and 5 (red) are markets.
- The minimum distance routes are:
  - 0 (Office) → 2 (Market) → 4 → 1 → 3 (Home), with a total distance of 4.
  - 0 (Office) → 2 → 4 (Market) → 1 → 3 (Home), with a total distance of 4.
  - 0 (Office) → 1 → 4 (Market) → 1 → 3 (Home), with a total distance of 4.

4 Skyline

4.1 Problem Statement

Consider a 2D plane with the X and Y axes as the horizontal and vertical axes. You are given N rectangles on this plane, each of which has its base on the x-axis.
Your task is to output the skyline formed by these N rectangles, which is the boundary visible when these rectangles are viewed along the Y-axis from top to bottom. Mathematically, the skyline is the set of all points $\{x_i, y_i\}$ such that the point lies in at least one rectangle and y_i is maximum for the given x_i. (Check the figure below for better understanding)
The expected time complexity for this task is $O(NlogN)$ where N is the number of rectangles.

4.2 Input Format

- The first line contains a single integer N, the number of rectangles.
- The next N lines contain three integers: $\langle \mathrm{L}_i, \mathrm{R}_i, \mathrm{H}_i \rangle$, which represent the leftmost x-coordinate, the rightmost x-coordinate, and the height of the rectangle, respectively.

4.3 Output Format

- Print a sequence of key points (x_i, y_i) representing the skyline. Each key point is the left endpoint of some horizontal segment in the skyline, except the last point in the list, which always has a y-coordinate 0 and is used to mark the skyline's termination where the rightmost rectangle ends.
- Each key point is separated by a space.(Check the example for more details).
- Note : Any empty ground between the leftmost and rightmost rectangles should be part of the skyline's contour.

4.4 Constraints

- $1 \leq N \leq 10^4$
- $1 \leq L_i < R_i \leq 2^{31} - 1$
- $1 \leq H_i \leq 2^{31} - 1$

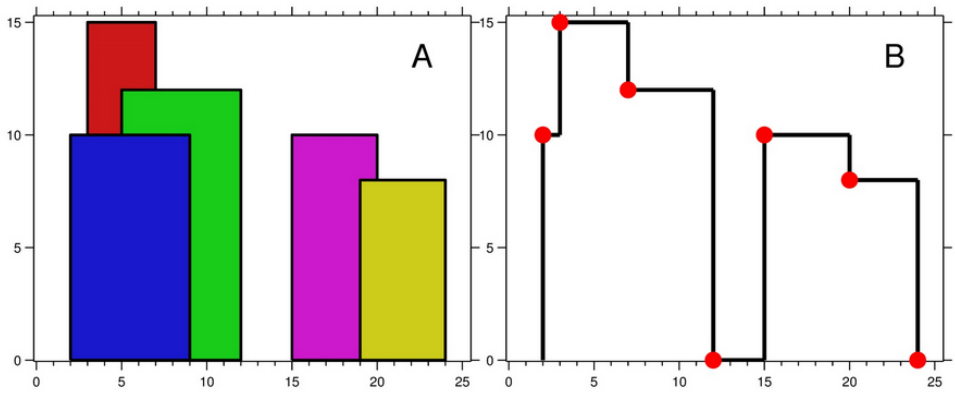
4.5 Example 1

Input:

```
5
2 9 10
3 7 15
5 12 12
15 20 10
19 24 8
```

Expected Output:

```
2 10 3 15 7 12 12 0 15 10 20 8 24 0
```



Explanation: Figure A shows the rectangles of the input. Figure B shows the skyline formed by these rectangles. The red points in figure B represent the key points in the output list.Note that the empty ground from x=12 to x=15 is also a part of the skyline's contour.

4.6 Example 2

Input:

```
3
1 3 4
4 6 5
7 9 3
```

Expected Output:

```
1 4 3 0 4 5 6 0 7 3 9 0
```

Test Cases

You will be evaluated on several hidden test cases. These test cases will not be revealed to you. For each hidden test case, you need to match the expected output. Then only will you get the full marks. You will not see your score until the deadline is over when we grade your code using autograder, and publish the result. You may resubmit as many times as you want, but only the final submission will be graded.