Assignment 4
CSC3100 Data Structures, Spring 2022

Due: 23:59, May 8, 2022

Hanzhe Wu

May 6, 2022

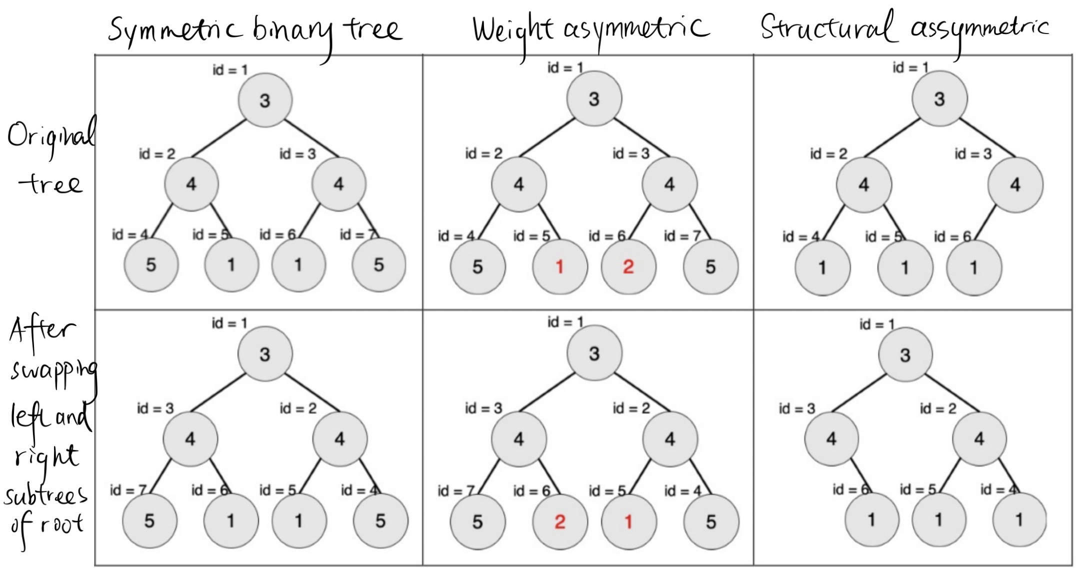
1 Problems

1.1 Problem 1: Symmetric Binary Tree

Description

WARNING: some definitions in this problem may be different with what you have learned in the lectures. Please follow the definitions below **in this problem**, while following what you have learned in exams, etc.
A rooted tree with weights in each node is called a *symmetric binary tree* if (and only if)

- It is a binary tree.
- After swapping the left and the right subtrees of **all** the nodes in the tree, the structure of the tree is identical to the original tree, and the weights in each corresponding positions are also equal to those of the original tree.



Please note that for the first column, you need to swap the left and right subtrees of **each** of the 7 nodes. In other words, you need to conduct 7 subtree swaps (counting swapping empty subtrees) in total.
Now given a binary tree, you are asked to find a subtree such that

- The subtree is a symmetric binary tree.
- The subtree has the maximum number of nodes.

Please output the **number of nodes** in the subtree.
Please note that a tree with only one node (i.e. its root) is a symmetric binary tree. We define a *subtree rooted at node T* to be the binary tree consisting of node T and **all** its decendants.

Input

- Line 1: One positive integer n, representing the number of nodes in the given tree. The nodes in the tree are indexed $1, \cdots, n$, where the node 1 is the root.
- Line 2: n positive integers separated by one space. The i-th integer v_i represents the weight in node i.
- For the following n lines, there are two positive integers l_i, r_i in the i-th line (of the n lines), separated by one space, representing the indices of the left and the right child of node i, respectively. If there is no left or right child for node i, then the corresponding integer is -1.

Output

- Line 1: One integer, representing the number of nodes in the maximal symmetric binary subtree of the given tree.

Sample I

Sample Input

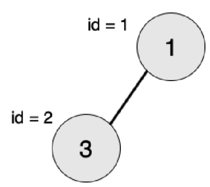
```
2
1 3
2 -1
-1 -1
```

Sample Output

```
1
```

Explanation

The maximal symmetric binary subtree is the subtree rooted at node 2, and the number of node(s) is 1.



Sample II

Sample Input

```
10
2 2 5 5 5 5 4 4 2 3
9 10
```

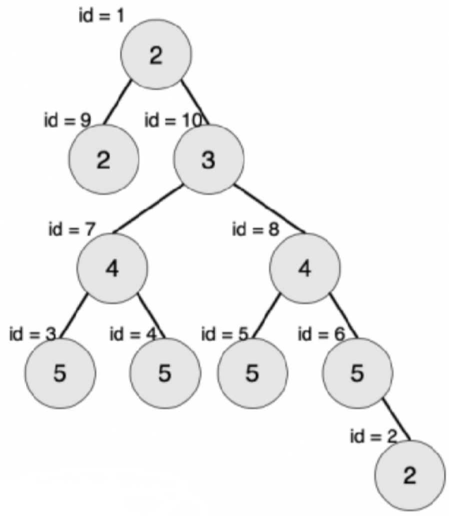
```
-1 -1
-1 -1
-1 -1
-1 -1
-1 2
3 4
5 6
-1 -1
7 8
```

Sample Output

```
3
```

Explanation

The maximal symmetric binary subtree is the subtree rooted at node 7, and the number of node(s) is 3.



Limits

- $1 \leq v_i \leq 1000, \forall i$
- $1 \leq n \leq 10^6$

Range of data

There are 25 test points in total, each accounting for 0.8% of the whole grade.

- For test points $1 \sim 3$, $n \leq 10$, it is guaranteed that all the nodes in the left subtree of the root node has no left child, and all the nodes in the right subtree of the root node has no left child.
- For test points $4 \sim 8$, $n \leq 10$.
- For test points $9 \sim 12$, $n \leq 10^5$, and it is guaranteed that the input represents a *full binary tree*(Please see the definition in the "Remarks").
- For test points $13 \sim 16$, $n \leq 10^5$, and it is guaranteed that the input represents a *complete binary tree* (Please also see the definition in the "Remarks").

- For test points $17 \sim 20$, $n \leq 10^5$, and it is guaranteed that the weight of all nodes of the inputted tree is all 1.
- For test points $21 \sim 25$, $n \leq 10^6$.

Remarks

We use the following definitions in this problem:

- **Level**: The level of the root node is 1, and the level of the children of the root is 2. For any node in the tree, the level of the node is defined as the level of its parent node plus one.
- **Depth of the tree**: The maximum level of the nodes in the tree is called the depth of the tree.
- **Full binary tree**: If the depth of a binary tree is h, and there are $2^h - 1$ nodes in the tree, then that tree is a full binary tree.

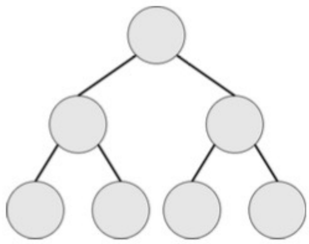


Figure 1: A full binary tree with depth 3

- **Complete binary tree**: If the depth of a binary tree is h, and all the levels except the h-th level reach their maximum size, and all the nodes in level h are continuously arranged from left to right (similar to the tree in heaps), then the tree is a complete binary tree.

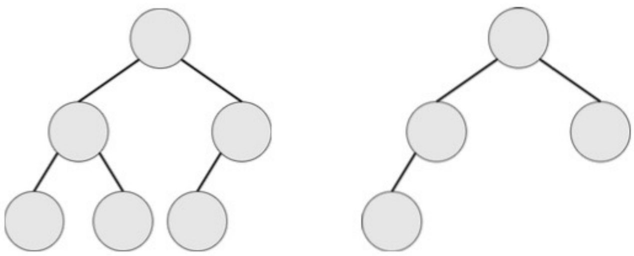


Figure 2: Two complete binary trees with depth 3

1.2 Problem 2: Non-repeating Numbers

Description

Prof. W gives you n numbers. She asks you to remove the repeated ones among them, leaving the one that **appeared first**.

Input

There are *multiple* dataset in each test. Previous datasets has no effect on new datasets.

- Line 1: one integer T, representing the number of datasets in the test.

For each dataset,

- Line 1: one integer n.
- Line 2: n integers representing the integers given.

Output

For each dataset, print one line of numbers, representing the remaining numbers after removal of repeated numbers. Two numbers are separated by exactly one space. The space after the last element in each row and the extra line after all n rows would be automatically ignored by the grader.

Limits

- $1 \leq T \leq 50$
- $1 \leq n \leq 5 \times 10^4$
- The numbers given is in 32-bit **signed** integers, i.e. $[-2^{31}, 2^{31} - 1]$.

Range of data

- For 30% data, $n \leq 100$, numbers given $\in [0, 100]$.
- For 60% data, $n \leq 10^4$, numbers given $\in [0, 10^4]$.
- For 100% data, $n \leq 5 \times 10^4$, numbers given $\in [-2^{31}, 2^{31} - 1]$.

Remark

WARNING: There are *multiple* dataset in each test. Please remember to clear up all the intermediate variables before entering a new set of data, or it may cause unprecedented problems.

Sample I

Sample Input

```
2
12
1 12 8 3 3 9 12 3 10 8 5 7
6
1 2 3 4 5 6
```

Sample Output

```
1 12 8 3 9 10 5 7
1 2 3 4 5 6
```

Sample II

Sample Input

```
1
8
1 9 2 6 0 8 1 7
```

Sample Output

```
1 9 2 6 0 8 7
```

1.3 Problem 3: Wandering

Description

Prof. X loves to wander around LGU before he returned to his big residental villa everyday. However, today he needs to go home more quickly. Although he wanted a fast trip home, he did not want to go along the shortest path, since it would not be interesting. He decided to choose the second shortest path.
LGU consists of R **bidirectional** roads, each linking two of the N intersections(vertices) numbered $1, 2, \cdots, N$. Prof. X starts at intersection 1, and his villa is at intersection(vertex) N.
Please note that

- The second shortest path may share roads with any of the shortest paths, and it may backtrack, i.e., use the same road or intersection(vertex) more than once.
- The second shortest path is the shortest path whose length is **strictly longer** than the shortest path(s) (i.e., if two or more shortest paths exist, the second-shortest path is the one whose length is longer than those but no longer than any other path).

Input

- Line 1: Two space-separated integers N and R
- Line 2 to $R+1$: The $(i+1)$-th line contains three space-separated integers: A_i, B_i, and D_i, describing a **bidirectional** road that connects intersections A_i and B_i and has length D_i.

Output

- Line 1: The length of the second shortest path between node 1 and node N.

Limits

- $1 \leq N \leq 5000$
- $1 \leq R \leq 10^5$
- $1 \leq D_i \leq 5000$, $\forall i = 1, \cdots, N$

Range of data

- For 100% data, $1 \leq N \leq 5000, 1 \leq R \leq 10^5, 1 \leq D_i \leq 5000$, $\forall i = 1, \cdots, N$

Sample I

Sample Input

```
4 4
1 2 100
2 4 200
2 3 250
3 4 100
```

Sample Output

```
450
```

Explanation

There are two routes: $1 \to 2 \to 4$ (with length $100 + 200 = 300$) and $1 \to 2 \to 3 \to 4$ (with length $100 + 250 + 100 = 450$)

Sample II

Sample Input

```
4 5
1 2 100
2 4 200
2 3 100
3 4 100
1 4 301
```

Sample Output

```
301
```

Sample III

Sample Input

```
5 5
1 2 5
2 3 5
3 4 4
4 5 6
1 5 21
```

Sample Output

```
21
```

1.4 Problem 4: Pandemic

Description

In 2202, due to a new surge of interstellar pandemic, all the planets in LGU were isolated, and students and faculties cannot go from one planet to another. To return the campus to normal, the LGU governing board decided to use starships to travel between different planets.
There are N planets in LGU, and the i-th planet can be represented as a point (x_i, y_i) in the **2-dimensional** plane. The cost to build a **bidirectional** starship airline between planets i and j is equal to the **squared** Euclidean distance between them, i.e.

$$(x_i - x_j)^2 + (y_i - y_j)^2 \tag{1}$$

Notice that we treat the airlines as edges between vertices. It means the starships do not exchange passengers in the middle of the airline. Also, the planets are in a 2D-plane instead of 3D-space to simplify the calculation.
The board of LGU decided to connect **all** the planets by a network of airlines with the minimum cost. However, the interstellar government would only accept LGU to open the airline between two planets with **squared** Euclidean distance **greater than or equal to** a constant C, since they believe the airline between two close planets would cause diseases to spread severely in both planets.
We say a set of planets are **connected** if (and only if) for **each** pair of planets in the set, there exists a **path** (not necessarily an airline (edge)) from one planet to another.
Also, the airline is **bidirectional**, it means that the cost would only be calculated once for both directions of the airline.
You, as the Chief Executive Officer of LGU, are responsible for calculating the minimum cost to connect all the planets of LGU.

Input

- Line 1: Two integers N and C.
- Line 2 to $N + 1$: The $(i + 1)$-th line contains two integers x_i, y_i, representing the location of the i-th planet in xy-coordinate.

Output

- Line 1: The minimum cost of a network of airlines connecting all the planets, or -1 if no such network can be built.

Limits

- $1 \leq N \leq 2000$
- $0 \leq x_i, y_i \leq 1000$, $\forall i = 1, \cdots, N$
- $1 \leq C \leq 10^6$

Range of data

- For 40% data, $1 \leq N \leq 200, 0 \leq x_i, y_i \leq 100$, $\forall i = 1, \cdots, N, 1 \leq C \leq 500$.
- For 100% data, $1 \leq N \leq 2000, 0 \leq x_i, y_i \leq 1000$, $\forall i = 1, \cdots, N, 1 \leq C \leq 10^6$

Sample I

Sample Input

```
3 11
0 2
5 0
4 3
```

Sample Output

```
46
```

Explanation

Input

There are 3 planets at locations $(0, 2)$, $(5, 0)$, and $(4, 3)$. LGU can only open airlines of cost(squared Euclidean distance) at least 11.

Output

LGU cannot open an airline between the planets at $(4, 3)$ and $(5, 0)$, since its cost would be only 10. It therefore opens an airline between $(0, 2)$ and $(5, 0)$ at cost 29, and an airline between $(0, 2)$ and $(4, 3)$ at cost 17.

Sample II

Sample Input

```
7 77
7 4
10 3
7 13
16 3
1 10
8 10
11 13
```

Sample Output

```
583
```

Sample III

Sample Input

```
5 10
11 7
5 11
6 6
8 9
6 10
```

Sample Output

```
55
```

2 Requirements

You can write your code in Java, Python, C, or C++. The time limit may vary among different languages, depending on the performance of the languages. Your code must be a **complete, runnable** program instead of only several functions. We guarantee that the input data used to test strictly conforms to the requirements in the description, and you do not need to deal with cases where the input data is invalid. If you do not understand the meaning of the problems, you may refer to the sample input/output/explanations or ask TAs or USTFs.

You also need to write a report to explain the following:

- What does your algorithm do in your program?
- Why you use this way to solve the problem? In other words, why your algorithm is **correct** to solve the problem? (No need for formal proof)
- **(Only for problems 2 and 4)** What is the time complexity of your program and why?

Please note that the maximum number of pages allowed for your report is **14 pages**. Please note that you do NOT need to write a lot to meet the page limit. A clear, logical and succinct report that answer all of the questions is encouraged and preferred.

3 Reference Problem

Please refer to Assignment 2 for details.

4 Submission

4.1 Online Judge

Please refer to Assignment 2 for details.
You can contact the platform administrator Hu Haichuan (`119010103@link.cuhk.edu.cn`) for issues related to the Online Judge system.

4.2 Blackboard

You also need to upload **both your code and report** to the Blackboard platform. You need to name each of your files according to the following rules and compress them into a `.zip` archive, where *ID* is your student ID.

```
A4_Submission_ID.zip
├── A4_P1_ID.java/py/c/cpp/cc
├── A4_P2_ID.java/py/c/cpp/cc
├── A4_P3_ID.java/py/c/cpp/cc
├── A4_P4_ID.java/py/c/cpp/cc
└── A4_Report_ID.pdf
```

5 Grading

- Code (90%): strictly dependent on your score on the Online Judge platform
  - Problem 1 (20%), Problem 2 (20%), Problem 3 (30%), Problem 4 (20%)
- Report (10%)

6 Remarks

If you have any question for this assignment, please contact Wu Hanzhe (`120090353@link.cuhk.edu.cn`).
Please note that the problems and test data may subject to minor change after releasing, especially in the first few days. Please keep track of the updates on the assignment, if any.
The problemset is somewhat challenging, so do not wait until the last minute to finish it.