CSE221
LabAssignment07
Spring2023

Submission Guidelines

1. You can code all of them either in Python, CPP, or Java. But you should choose a specific language for all tasks.
2. For each task write separate python files like task2.py, task3.py, and so on. For problems that have subproblems, name those like task1A.py, task1B.py, and so on.
3. For each problem, take input from files called **"inputX_Y.txt"** and output at **"outputX_Y.txt"**, where X is the task number and Y is the sample i/o number. For example, for problem 2 sample 1, the input file is this, "input2_1.txt". For problems that have subproblems, name the files like "input1a_1.txt", "output1a_1.txt" and so on. Same for output.
4. For each task include at least one input file (if any) in the submission folder.
5. Finally zip all the files and rename this zip file as per this format:**LabSectionNo_ID_CSE221LabAssignmentNo_Spring2023.zip** [Example:**LabSection01_21101XXX_CSE221LabAssignment05_Spring2023.zip**]
6. Don't copy from your friends.
7. You MUST follow all the guidelines, naming/file/zipping convention stated above.
   Failure to follow instructions will result in a straight 50% mark deduction.

Task 1 [10 Marks]

You are a busy person with lots of tasks to do. You have a schedule of tasks represented by intervals of time, where each interval represents a task that you need to complete. However, you can only work on one task at a time, and you want to complete as many tasks as possible.

Given a list of N intervals of time, your task is to determine the maximum number of tasks you can complete and which tasks they are.

Input

The input consists of a single integer N ($1 \leq N \leq 10^5$), the number of tasks, followed by N lines representing the tasks. Each task is represented by two integers S_i and E_i ($0 \leq S_i \leq E_i \leq 10^9$), the start and end times of the task, respectively.

Output

Output a single integer k, the maximum number of tasks you can complete, followed by a line with k intervals of the tasks you can complete.

If there are multiple solutions with the same maximum number of tasks, print any one of them.

Sample Input/Output:

Sample Input 1	Sample Output 1
6 1 3 2 5 3 7 4 6 6 8 7 9	3 1 3 4 6 6 8
Sample Input 2	Sample Output 2
5 1 4 2 5 6 7 4 8	2 1 4 6 7

3 6	
Sample Input 3	Sample Output 3
7 0 4 3 4 1 5 9 10 6 9 2 3 1 2	5 1 2 2 3 3 4 6 9 9 10

Task 2 [10 Marks]

Given N tasks and M people, where each task has a start time and end time, implement a greedy algorithm to find the maximum number of tasks that can be completed by M people.

Each task can only be completed by one person and a person can only be assigned one task at a time. Two tasks cannot be completed simultaneously by the same person.

Input

The input consists of two integers N and M ($1 \leq N, M \leq 10^3$), the number of activities and the number of people, respectively. This is followed by N lines representing the activities. Each line contains two integers S_i and E_i ($0 \leq S_i \leq E_i \leq 10^9$), representing the start and end times of the activity, respectively.

Output

Output a single integer representing the maximum number of activities that can be completed.

Sample Input/Output:

Sample Input 1	Sample Output 1
5 2 1 5 3 6 2 5 8 10 6 9	4
Sample Input 2	Sample Output 2
5 2 1 4 2 5 6 7 4 8 3 6	4
Sample Input 3	Sample Output 3
5 2 1 10 2 10 6 7 4 8 3 6	3
Sample Input 4	Sample Output 4
8 3 5 7 2 4 6 8 8 10 1 3 7 9 3 5 2 6	8

Sample Input Explanation:

In sample input 2-

Person 1 will complete the tasks: 1-4, 4-8
Person 2 will complete the tasks: 2-5, 6-7

Task 3 [10 Marks]

Study material: http://www.shafaetsplanet.com/?p=763

There is a group of N people living in a small village. They live in their own house. Although they are all neighbors, they don't all know each other very well.

Each person in that village has their own unique identity - labeled with an integer value between 1 to N. Initially, the villagers don't have any friends. As time passes by, they begin to make friendship between themselves.

In this problem, you will be given a description of K friendships. You have to print an integer value which denotes the size of their friend circle.

Suppose, there are five people living in the village labeled with 1,2,3,4 and 5. Initially, the size of each friend circle is one, since no friendship has been created yet.
One day, person 1 and person 2 become friends. So the size of their friend circle becomes two. Next day, person 3 and person 4 become friends and the size of their friendship becomes two as well. After a few days, person 1 and person 4 become friends. Now the size of their friend circle becomes four consisting of persons 1,2,3 and 4.

Input Format:

The input consists of two integers, separated by a space, denoting the number of people in the village, N ($1 \le N \le 10^5$) and the number of queries that will follow, K ($1 \le K \le 10^5$).

The next K lines contain two integers A_i and B_i each ($1 \le A_i, B_i \le N$ and $A_i != B_i$), separated by a space, representing two people who have become friends as a result of the query.

Output Format:

For each query, output a single integer on a new line representing the size of the friend circle that the two people belong to after becoming friends.

Sample Input/Output:

Sample Input 1	Sample Output 1
5 3 1 2 3 4 1 4	2 2 4
Sample Input 2	Sample Output 2
8 7 2 4 4 5 3 6 4 7 3 1 2 7 6 2	2 3 2 4 3 4 7

Sample Input Explanation:

In sample input 2,

Query 0: Initially, there are 8 people in the village who do not know each other.

{1} {2} {3} {4} {5} {6} {7} {8}

Query 1: After person 2 and person 4 becoming friends:

{1} {2,4} {3} {5} {6} {7} {8}

The output is 2, since the size of the friends circle {2,4} is 2.

Query 2: After person 4 and person 5 becoming friends:

{1} {2,4,5} {3} {6} {7} {8}

The output is 3, since the size of the friends circle {2,4,5} is 3.

Query 3: After person 3 and person 6 becoming friends:

{1} {2,4,5} {3,6} {7} {8}

The output is 2, since the size of the friends circle {3,6} is 2.

Query 4: After person 4 and person 7 becoming friends:

{1} {2,4,5,7} {3,6} {8}

The output is 4, since the size of the friends circle {2,4,5,7} is 4.

Query 5: After person 3 and person 1 becoming friends:

{2,4,5,7} {1,3,6} {8}

The output is 3, since the size of the friends circle {1,3,6} is 3.

Query 6: Since the person 2 and person 7 are already in the same friend circle, nothing changes:

{2,4,5,7} {1,3,6} {8}

The output is 4, since the size of the friends circle {2,4,5,7} is 4.

Query 7: After person 6 and person 2 becoming friends:

{1,2,3,4,5,6} {7} {8}

The output is 7, since the size of the friends circle {1,2,3,4,5,6} is 7.