CSC11F: Advanced Data Structures and Algorithms

Review

Yutaka Watanobe (142-A, yutaka@u-aizu.ac.jp)

1 Introduction

To undertake the course for advanced data structure and algorithm and solve the problems given, students should have knowledge and skills to understand/implement the following items:

- **Complexity**. Time and space complexity, Big-Oh notation, etc.
- **Basic Sort**. Bubble Sort, Selection Sort, Stability of sorting algorithms, etc.
- **Elementary Data Structures**. Stack, Queue, List, etc.
- **Search**. Linear Search, Binary Search, etc.
- **Divide and Conquer**. Recursion, Marge Sort, etc.
- **Advanced Sort**. Quick Sort, Counting Sort, etc.
- **Tree**. Tree Structure, traversing algorithm, etc.
- **Binary Search Tree**. Search, Insertion, Deletion, etc.
- **Heaps**. Building heaps, Priority Queue, Heap Sort, etc.
- **Dynamic Programming**. Longest Common Subsequence, Knapsack Problem, Matrix Chain Multiplication, etc.
- **Graph**. Graph representation, DFS, BFS, Connected Components, etc.
- **Weighted Graph**. Single Source Shortest Path, Minimum Spanning Tree, etc.
- **Heuristic Search**. A*, IDA*, etc.
- **String Matching**. Brute Force, KMP, BM method, etc.
- **Randomized Algorithms**. Random number generator, etc.

Actually, these fundamental topics and keywords have been covered by Algorithms and Data Structures I and II at undergraduate courses of the University of Aizu.

For the preparation for this advanced course, this week, students should try to solve some basic problems so that the instructor can understand your programming skill, experience and knowledge.

Note that you do not need to make excessive effort. If majority of students could not solve a certain problem, the corresponding algorithm can be a topic for the lecture.

2 Assignments

Place

https://onlinejudge.u-aizu.ac.jp/beta/room.html#CSC11F_2023_Week_01

Duration

1 week

Overview

#	Title	Note
A	Intersection of Circles	Check basic programming skill. You must solve this problem.
B	Cycle Detection for a Directed Graph	Check basic algorithmic skill. You should solve this problem.
C	Reconstruction of a Tree	Check basic algorithmic skill. You should solve this problem.
*D	15 Puzzle	If you can solve the problem without studying, then try. You do not need to make exessive effort to solve these problems. They can be topics of this course.
*E	Convex Hull	
*F	Bipartite Matching	

* option

Problem A: Intersection of Circles

For given two circles c_1 and c_2, print `4` if they do not cross (there are 4 common tangent lines), `3` if they are circumscribed (there are 3 common tangent lines), `2` if they intersect (there are 2 common tangent lines), `1` if a circle is inscribed in another (there are 1 common tangent line), `0` if a circle includes another (there is no common tangent line).

Input

Coordinates and radii of c_1 and c_2 are given in the following format:
cx_1 cy_1 r_1
cx_2 cy_2 r_2

Output

Print "`4`", "`3`", "`2`", "`1`" or "`0`" in a line.

Sample Input 1

```
1 1 1
6 2 2
```

Sample Output 1

```
4
```

Note

This is an easy problem to check your "programming skill". If you can not solve this problem, there is no chance to solve problems given in this core course.

Problem B: Cycle Detection for a Directed Graph

Find a cycle in a directed graph $G(V, E)$.

Input

A directed graph G is given in the following format:

$|V|$ $|E|$
s_0 t_0
s_1 t_1
:
$s_{|E|-1}$ $t_{|E|-1}$

$|V|$ is the number of nodes and $|E|$ is the number of edges in the graph. The graph nodes are named with the numbers $0, 1, ..., |V|-1$ respectively.

s_i and t_i represent source and target nodes of i-th edge (directed).

Output

Print `1` if G has cycle(s), `0` otherwise.

Constraints

- $1 \leq |V| \leq 100$
- $0 \leq |E| \leq 1,000$
- $s_i \neq t_i$

Sample Input 1

```
3 3
0 1
0 2
1 2
```

Sample Output 1

```
0
```

Sample Input 2

```
3 3
0 1
1 2
2 0
```

Sample Output 2

```
1
```

Note

This is a problem to check your knowledge of fundamental algorithms. You should try to apply a basic graph algorithm to solve this problem. Please try to solve this problem using a recursive function (That is depth-first search).

Problem C: Reconstruction of a Tree

Write a program which reads two sequences of nodes obtained by the pre-order tree walk and the in-order tree walk on a binary tree respectively, and prints a sequence of the nodes obtained by the post-order tree walk on the binary tree.

Input

In the first line, an integer n, which is the number of nodes in the binary tree, is given.

In the second line, the sequence of node IDs obtained by the pre-order tree walk is given separated by space characters.

In the second line, the sequence of node IDs obtained by the in-order tree walk is given separated by space characters.

Every node has a unique ID from 1 to n. Note that the root does not always correspond to 1.

Output

Print the sequence of node IDs obtained by the postorder tree walk in a line. Put a single space character between adjacent IDs.

Constraints

- $1 \leq n \leq 40$

Sample Input 1

```
5
1 2 3 4 5
3 2 4 1 5
```

Sample Output 1

```
3 4 2 5 1
```

Sample Input 2

```
4
1 2 3 4
1 2 3 4
```

Sample Output 2

```
4 3 2 1
```

Note

This is a good problem to review the pre-order, in-order and post-order tree traversal algorithms which are very important to implement advanced data structures.

Problem D: 15 Puzzle

The goal of the 15 puzzle problem is to complete pieces on 4×4 cells where one of the cells is empty space.

In this problem, the space is represented by 0 and pieces are represented by integers from 1 to 15 as shown below.

```
1 2 3 4
6 7 8 0
5 10 11 12
9 13 14 15
```

You can move a piece toward the empty space at one step. Your goal is to make the pieces the following configuration in the shortest move (fewest steps).

```
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 0
```

Write a program which reads an initial state of the puzzle and prints the fewest steps to solve the puzzle.

Input

The 4×4 integers denoting the pieces or space are given.

Output

Print the fewest steps in a line.

2.1 Constraints

- The given puzzle is solvable in at most 45 steps.

Sample Input

```
1 2 3 4
6 7 8 0
5 10 11 12
9 13 14 15
```

Sample Output

```
8
```

Note

This is an option. It can be a course topic of Heuristic Search.

Problem E: Convex Hull

Find the convex hull of a given set of points P. In other words, find the smallest convex polygon containing all the points of P. Here, in a convex polygon, all interior angles are less than or equal to 180 degrees.

Please note that you should find all the points of P on both corner and boundary of the convex polygon.

Input

n
$x_1\ y_1$
$x_2\ y_2$
:
$x_n\ y_n$

The first integer n is the number of points in P. The coordinate of the i-th point p_i is given by two integers x_i and y_i.

Output

In the first line, print the number of points on the corner/boundary of the convex polygon. In the following lines, print xy coordinates of the set of points. The coordinates should be given in the order of counter-clockwise visit of them starting from the point in P with the minimum y-coordinate, or the leftmost such point in case of a tie.

Constraints

- $3 \leq n \leq 100000$
- $-10000 \leq x_i, y_i \leq 10000$
- No point in the P will occur more than once.

Sample Input 1

```
7
2 1
0 0
1 2
2 2
4 2
1 3
3 3
```

Sample Output 1

```
5
0 0
2 1
4 2
3 3
1 3
```

Note

This is an option. It can be a course topic of Computational Geometry.

Problem F: Bipartite Matching

A bipartite graph $G = (V, E)$ is a graph in which the vertex set V can be divided into two disjoint subsets X and Y such that every edge $e \in E$ has one end point in X and the other end point in Y.

A matching M is a subset of edges such that each node in V appears in at most one edge in M.

Given a bipartite graph, find the size of the matching which has the largest size.

Input

$|X|$ $|Y|$ $|E|$
x_0 y_0
x_1 y_1
:
$x_{|E|-1}$ $y_{|E|-1}$

$|X|$ and $|Y|$ are the number of vertices in X and Y respectively, and $|E|$ is the number of edges in the graph G. The vertices in X are named with the numbers 0, 1,..., $|X|$-1, and vertices in Y are named with the numbers 0, 1,..., $|Y|$-1, respectively.

x_i and y_i are the vertex numbers from X and Y respectevely which represent the end-points of the i-th edge.

Output

Print the largest size of the matching.

Constraints

- $1 \leq X \leq 100$
- $1 \leq Y \leq 100$
- $0 \leq |E| \leq 10000$

Sample Input 1

```
3 4 6
0 0
0 2
0 3
1 1
2 1
2 3
```

Sample Output 1

```
3
```

Note

This is an option. It can be a course topic of Network Flow.