

10054 The Necklace

Set up

It was written in C++ 11. Run by command

```
g++ -lm -lcrypt -O2 -std=c++11 -pipe -DONLINE_JUDGE
```

It has been accepted by Uva.

Algorithm Overview

Assume we can treat each beads as an undirected edge by connecting two vertices. Then this question is able to convert to a typical question of printing the Euler tour. An *Euler path* is defined as a path in a graph which visits *each edge* of the graph *exactly once*. An *Euler tour* is an Euler path which starts and ends on the same vertex. In this case, if it is a valid necklace, then colors should be marked exactly once and the first value and last value have the same value. According to our textbook “Competitive Programming 3”, to check if a connected undirected graph has an Euler tour, we need to check if all its vertices have even degrees. It can be done in $O(|V| + |E|)$.

10147 Highways

Set up

It was written in **C++ 11**. Run by command

```
g++ -std=c++11 -lm -lcrypt -O2 -pipe -DONLINE_JUDGE
```

It has been accepted by Uva.

Graph building and MST algorithm was copied from code book

Algorithm Overview

- Built a graph
- **For existing highways, distance(cost) is 0**
- Find MST
 - Use kruskal

Trick

- Use inline for calculating distance
 - skip “sqrt” in distance since we are only comparing distance, sqrt is useless
-

589 Pushing Boxes

Set up

It was written in **C++ 11**. Run by command

```
g++ -std=c++11 -lm -lcrypt -O2 -pipe -DONLINE_JUDGE
```

It has been accepted by Uva.

BFS algorithm used in this problem is copied from code book.

Algorithm Overview

- Read in the user input and put them into a 2D array which acts as a “maze”. Also record the starting point of user, the box and the end point of the box.
 - Start the BFS process for the box from the starting point location.
 - Inside the box’s BFS process, we need to do BFS for the person to ensure that the person can reach the box.
 - We will check whether the box can be moved to 4 directions: up, down, left and right.
-

208 Fire Truck

Set up

It was written in **Java**. Compile by command

```
javac P208.java
```

It has been accepted by Uva.

If you submit our Java source code to UVa, please change the class name which contains the main function to `'Main'`, otherwise UVa will report compilation error.

BFS and DFS algorithm used in this problem is copied from code book.

Algorithm Overview

- Read in user input and construct a graph containing cities connected.
- Apply BFS to the graph to check whether the destination is connected to 1. This check will eliminate the effort to generate a path to the destinations where there is no way to go to from the source vertex 1.
- After we found out that the given destination is connected to vertex 1, we can apply DFS with source 1 and the given destination to generate a path.
- After that, we can output the path or “No route to destination” based on the path result returned from previous step.