Cairo University
Faculty of Engineering
Computer Engineering
Fall 2019

Design and Analysis of Algorithms
Homework 1
Brute Force Algorithms

Requirements

1. Maximum Clique Problem

Given an undirected graph $\boldsymbol{G = (V, E)}$ where $\boldsymbol{V}$ is the set of vertices and $\boldsymbol{E}$ is the set of edges. A clique $\boldsymbol{V'}$ is a set of vertices subset of $\boldsymbol{V}$ where each vertex in $\boldsymbol{V'}$ is connected to every other vertex in the set with an edge. A maximum clique is the clique with the largest number of vertices possible.

You are required given a graph $\boldsymbol{G = (V, E)}$ to output maximum clique size using **non-optimized brute force** solution.

The input format will be as follows:

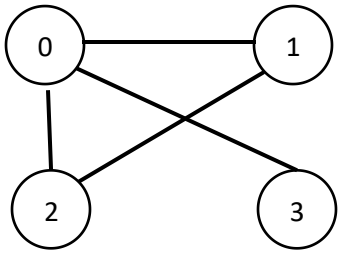
- The first line will contain the number of vertices of the graph $\boldsymbol{V}$.
- The second line will contain the number of edges in the graph $\boldsymbol{E}$.
- The next $\boldsymbol{E}$ lines will contain an edge each. Each line will contain two space-separated indices representing the vertices connected with an edge.

Example Input:

```
4
4
0 1
0 2
```

0 3
1 2

This input corresponds to the following graph:



You should output:

- the size of the maximum clique in one line

Example output for the given input:

3

2. Travelling Salesman Problem

Given a set of cities and the distance between each two cities, the goal of the travelling salesman problem is to find the shortest route to visit every city exactly once and return to the starting city. The problem can be perceived as a weighted undirected graph, where cities are vertices and distances between cities are weighted edges.

You are required given a set of cities and the distance between them to output the cost of the minimum route satisfying the definition described above using **non-optimized brute force** solution.

The input format will be as follows:

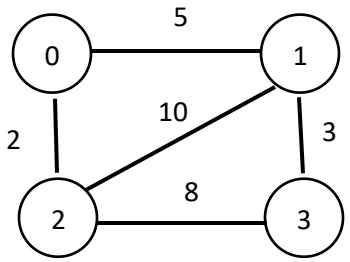
- The first line will contain the number of cities followed by the source city index.
- The second line will contain the number of edges between cities ***E***.

- The next ***E*** lines will contain an edge each. Each line will contain three space-separated numbers. The first two represents the cities connected with the edge. The last number represents the weight of the edge (distance)

Example Input:

4 0
5
0 1 5
0 2 2
1 2 10
1 3 3
2 3 8

This input corresponds to the following graph:



You should output:

- The cost of the trip in one line **-1** if no route found.

Example output for the given input:

18

3. Matrix Multiplication Order

Given a chain of matrices, find the most efficient order in which to do the multiplication for minimum operations using **non-optimized brute force** solution.

Matrix multiplication is associative. This means whatever the parentheses order you use the results is the same (ABC)D = (AB)(CD) = A(BCD). For example, (AB)C = A(BC). However, if A is a 5*3 matrix, B is a 3*2 matrix, and C is a 2*10, then

- (AB)C yields 130 multiplications. (5*3*2) + (5*2*10)
- A(BC) yields 210 multiplications. (3*2*10) + (5*3*10)

The input format will be as follows:

- One line containing the chain of matrix order.

Examples input:

5 3 2 10

This means first matrix is of size 5*3, second matrix is 3*2, and third matrix is 2*10.

You should output:

- One number indicating the minimum number of operations possible.

Example output for the given input:

130

Submission Requirements

- You are required to create an account on https://www.spoj.com.
- Fill in the form (will be sent to email) with your name and your username on spoj.
- You are required to submit your C++ code to the contest (will be sent to email). Make sure that your solution is accepted by the judge.
- Prepare a report file with the following:
    - Locally on your computer run your solution code over different test cases containing different values of input size and measure the execution time (without reading input and writing output time).

- Plot execution time vs input size for each problem and include it in the report
- Comment on the results
- The next table will contain input size required for each problem and assumptions (You can stop at an smaller input size if running is taking more than 5 minutes for one test case)

Problem	Maximum Clique	Travelling Salesman Problem	Matrix Multiplication Order
Input Size	Vary the number of vertices from 2 to 30	Vary the number of cities from 3 to 14	Vary the number of matrices from 3 to 20
Assumptions	Use a fully connected graph for each test case	Use a fully connected graph with a fixed weight for each edge. For example, the distance between each city is (5)	Use a single number for all matrix sizes.