

Design and Analysis of Algorithms
Lab 1
Brute Force Algorithms

Objectives

After this lab, the student should be able to:

- handle problem requirements and generate correct output for any given input.
- provide non-optimized brute force solution.

Requirements

1. Vertex Cover Problem

Given an undirected graph $G = (V, E)$ where V is the set of vertices and E is the set of edges. A vertex cover V' is a set of vertices subset of V where for each edge $e = (u, v)$ either vertex u or v are in V' . The vertex cover problem is to find the minimum size vertex cover for a given undirected graph.

You are required given a graph $G = (V, E)$ to output minimum vertex cover using **non-optimized brute force** solution.

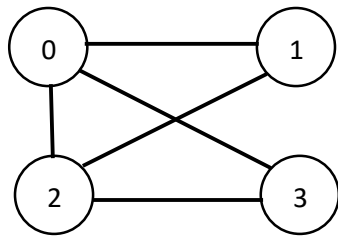
The input format will be as follows:

- The first line will contain the number of vertices of the graph V .
- The second line will contain the number of edges in the graph E .
- The next 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
5
0 1
0 2
0 3
1 2
2 3
```

This input corresponds to the following graph:



You should output

- the size of the minimum vertex cover set in one line,
- the minimum vertex cover set in a separate line.

Example output for the given input:

```
2
0 2
```

Here, the minimum set size is 2 and there is only one set satisfying the requirement which consists of vertices 0 and 2.

Note that $\{0, 1, 3\}$ is a vertex cover but is not minimal.

2. Subset Sum Problem

Given a set of numbers N and a value V , find a subset of N that have a summation equal to V using **non-optimized brute force** solution.

The input format will be as follows:

- The first line will contain the numbers of set N space-separated.
- The second line will contain the value V .

Example Input:

```
8 10 12 25
35
```

You should output:

- 0 or 1 in one line. 0 indicating no possible sets, and 1 indicating there is at least one set with the required sum.
- then output one set satisfying the sum in a separate line.

Example output for the given input:

```
1
10 25
```