CSCI 406: AlgoBOWL

Reminder: You are NOT allowed to consult the internet to solve this problem.

1 Problem Description

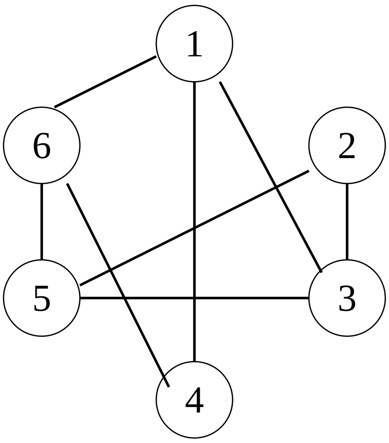
Your input is a weighted undirected graph G with an even number of vertices and a set of edges connecting pairs of distinct vertices. The objective is to design an algorithm that splits the vertices into two sets of *equal* size so that the sum of the weights of edges crossing between vertices in different sets is minimized.

Input Format:

The input will be provided in a file in the following format: the first line contains two integers: the first denotes n, the number of vertices and the second denotes m, the number of edges in the graph (n should be even). The vertices of the graph may then be denoted by integers in the range $[1, n]$. Each subsequent line in the graph denotes an edge in the graph followed by its weight.

```
6 8
1 3 1
1 4 1
1 6 1
2 3 1
2 5 1
3 5 1
4 6 1
5 6 1
```

The input above describes a graph with 6 vertices labeled 1 through 6 and eight undirected edges each with weight 1. The first of these edges joins vertices 1 and 3. The resulting graph is shown below. Observe that a split of ({1,4,6}, {2,3,5}) results in weight 2, whereas a split of ({1,2,3}, {4,5,6}) results in weight 4, making the first split superior to the second.



Input Restrictions: No input will contain more than 1,000 vertices and 100,000 edges. Edge weights are integers in [1,50].

Output Format: Line 1 of your output will contain the weight of your solution. Line 2 will contain vertex IDs assigned to the first set and Line 3 will contain vertex IDs assigned to the second set. To illustrate, the output for the second split in the example is

```
4
1 2 3
4 5 6
```

Note: This problem is NP-hard, which means that it is unrealistic to expect that your algorithm will compute an optimal solution in a reasonable time frame. Please keep this in mind as you work on this project.

2 Deliverables

Your group has three tasks:

1. Develop as good an algorithm as you can that accepts a valid input and produces a valid output.
2. Create an input within the parameters specified above that will challenge the other groups.
3. Develop a tool that verifies the other groups' outputs. What does this mean? The purpose of the verifier is to examine the outputs that other groups compute on your input and check that the overall weight they are reporting is consistent with their distribution of vertices into sets. (The verifier is not checking whether the solution provided by the other group is optimal.)

3 Grades

Overview: AlgoBOWL is 10% of your total grade. Your grade will be based on the following:

1. Does your software work and produce correct (i.e., valid, not optimal) solutions in a reasonable amount of time? [$\approx$ 75% of the grade].
2. How good were your inputs at stress-testing the other groups' algorithms? [$\approx$ 10% of the grade]. An incorrectly formatted input or one that does not adhere to the specs can result in your group forfeiting these points!
3. How do the solutions produced by your algorithm compare to those of other groups? [remaining $\approx$ 15%].
4. Your effort distribution within the group.

4 Logistics

You will use `algobowl.mines.edu` for the three steps below.

1. Step 1: Upload your group's input by 6pm on Wed, Feb 21.
2. Step 2: Upload your group's output on *each of the other group's inputs* by 6pm on Thu, Feb 22.
3. Step 3: Verify all of the other group's outputs on your input by 6pm on Fri, Feb 23.
4. Step 4: Upload your group's effort distribution by 6pm on Fri, Feb 23.

Good Luck!!!