Institut für Informatik
Prof. Dr. Heiko Röglin,
Dr. Daniel Schmidt

Lab: Efficient Algorithms
for selected Problems
Winter 2019/20

Problem Set 5

1 k-island clustering

Task: In this task, we consider a clustering problem that is a maximization problem. We call it the *k-island clustering* problem. Given are n points P in a metric space, and a number $k \geq 2$. The goal is to find a clustering with k clusters that *maximizes* the *minimum* distance between any two clusters in the partitioning, i.e., a partitioning $(P_i)_{i=1}^k$ that maximizes

$$\min_{i,j\in[k]} \min_{x\in P_i, y\in P_j} d(x,y).$$

The distances are given by a graph metric, i.e., the input contains an undirected weighted graph $G = (P, E)$, and the distance between $i, j \in P$ is the length of a shortest path between i and j in G. The input graph is connected, so all distances are finite.

Input: The first line contains n, the second line contains k. We adopt the names $0, \ldots, n-1$ for the n vertices. The third line contains m, the number of edges. Each edge is then given by a line containing three integers, the vertices i and j that it connects, and the (integral) weight w that the edge has. For every pair of vertices, there is at most one edge specified.

Output: The value of an optimum solution.

Sample Input:

```
10
2
12
0 1 1
0 4 5
1 2 1
2 3 1
3 4 1
4 5 1
5 6 1
6 7 1
7 8 2
8 9 1
1 3 6
6 9 4
```

Sample Output:

```
2
```

2 Fridays for Future 2

After the great results of the green party at the elections, people feel your goal has been achieved and interest in fridays for future demonstrations in your home town has dramatically decreased. All but six fellow protesters have deserted the movement. Is there a way to distribute the six protesters on road intersections in such a way that everyone that travels along a road will see at least one protester?

Input: Your home town is given as a graph where roads are represented by edges and intersections are represented by nodes. The first line of the input contains two numbers, the number of nodes n, and the number of edges m. This is followed by m lines containing two integers $a, b \in \{0, \ldots, n-1\}$. Each such pair describes an edge in the graph.

Output: Output `possible`, if you can find a suitable vertex cover with six vertices, otherwise output `impossible`.

Sample Input:

```
7 7
0 1
1 2
2 3
3 4
4 5
5 6
6 0
```

Sample Output:

```
possible
```

3 The Fast and the Furious II

Task: After your last race in Manhattan you appeared in the list of the most wanted people of the FBI. So you decide that it is better to escape from the United States and go to South America. Unfortunately, you also have to leave your loved Lamborghini behind.

The street racing competitions here are similar to the ones in Manhattan: You and your opponent are given some destinations, must visit them all and the one who returns to the start first wins. But there are also some differences: The street network here is different than in Manhattan, usually you have to visit fewer destinations, and your new car is not faster than your opponent's car.

Input: The first line contains three integers n, m and d. n is the number of crossroads, which are numbered from 0 to $n-1$, m is the number of streets, and d the number of destinations (including the starting point). Then, m lines follow, which describe the streets that connect the crossroads: Each line contains three values a_i, b_i and w_i. a_i and b_i are the two crossroads that are connected by this street, and w_i denotes its length. Finally, in a last line the d destinations are described. Here, the first destination is start (and finish) of the race.

You can assume that $1 \leq n \leq 50000$, $d \leq 20$, and all values are integers. Moreover, you can drive on the streets in both directions.

Output: Output the length of a shortest tour beginning and finishing in the start and visiting all destinations.

Sample Input:

```
9 12 3
0 1 2
0 3 5
1 2 4
1 4 2
2 5 2
3 4 5
3 6 1
4 5 4
4 7 4
5 8 2
6 7 3
7 8 5
0 2 8
```

Sample Output:

```
20
```

4 Skyline

Task: You are a famous architect and your current project is to plan an entire new city. Since you are impressed by Manhattan you decide that the streets will form a grid, and in each block there shall be a gigantic skyscraper. Investors already handed in plans for the buildings. You are interested in the minimum distance between two skyscrapers of the same height, because this is important to ensure that the city will have a nice skyline.

Input: The blocks of the city are arranged in a grid of dimension n. The first line of the input contains this number. Then n lines follow. The i-th line contains the n heights h_{ij} of the skyscrapers in row i.

You can assume without loss of generality that the heights are in $\{0, \ldots, n^2 - 1\}$ and that there are at least two skyscrapers of the same height. The distance between two blocks is given by the Manhattan distance.

Output: Output the minimum distance between two skyscrapers of the same height.

Sample Input 1:

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

Sample Output 1:

```
1
```

Sample Input 2:

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

Sample Output 2:

5 Lazy Teacher

Task: The students in the Lab are unhappy with the workload. Some ask for more difficult tasks while others want easier exercises. To accommodate the students, you decide to let them vote on how the next exercise sheet should look like. Towards this end you select a group of students that will form a committee of size k and vote on the matter. Every member of this committee has to either vote increase or decrease the workload.

What the students do not know is that you have already prepared the next tasks and because you are lazy, you prefer to not change anything about it. So the best outcome for you is if the vote ends in a tie. You know that every student $i \in [n]$ will vote to increase the workload with probability p_i and vote to decrease workload with probability $(1 - p_i)$.

Choose a subset of the students $S \subseteq N$ with $|S| = k$ such that the probability to get a tie is maximized.

Input: The first line contains the number of candidates n and the size of the committee k. Here, k is guaranteed to be even. After that, each line contains the probability p as a floating point number that the corresponding candidate votes to increase the workload.

Output: For each test case, the output is a single line containing the maximal probability to achieve a tie rounded to 3 decimal places.

Sample Input 1

```
4 2
0.4
0.8
0.3
0.5
```

Sample Output 1:

```
0.62
```

Sample Input 2

```
8 4
0.1
0.3
0.9
0.6
0.5
0.3
0.9
0.5
```

Sample Output 2:

```
0.572
```

6 Smooth Operator

You operate a large energy network consisting of sites and links between the sites. Each link has a small chance to fail. Since you mostly stare at old monochrome displays and state-of-the-art Windows NT applications, your thoughts drift off and you wonder about the reliability of the network. From historical data, you know the failure probability of each link in the network. You also know that the entire network breaks down if – as a result of link failures – the network splits into (at least) two disconnected parts. So you come up with the following idea to gauge the reliability: If you look at any subset S of the sites, then the probability that all links connecting sites in S to sites outside of S fail should be less than 10%. Can you figure out which of the networks you operate are reliable?

Input: The first line contains the number n of sites, the second line contains the number m of links. The following m lines contain two integers i and j, $i < j$, and a floating point number $p_{ij} \in (0, 1]$ with three places after the decimal point, thus specifying the failure probability p_{ij} of the link between site i and j. Links with zero error probability are left out, so you may assume that $p_{ij} > 0$. Sites are indexed from $0, \ldots, n-1$.

Output: Output `1` if the network is reliable and output `0` otherwise. The network is reliable if for all subsets S of sites it holds that the combined failure probability $\prod_{i \in S, j \notin S} p_{ij}$ is strictly less than 10%.

Sample input:

```
4
4
0 1 0.1
1 2 0.1
2 3 0.1
0 3 0.1
```

Sample output:

```
1
```

Sample input II:

```
4
4
0 1 0.1
1 2 0.1
2 3 0.3
0 3 0.4
```

Sample output II:

```
0
```