Algorithms Homework #3

Due: 2017/12/13 (Wed.) 03:00 (Programming)
2017/12/13 (Wed.) 14:20 (Hand-written)

Contact TAs: alg2017ta@gmail.com

Instructions

- There are two sections in this homework, including the hand-written part and the programming part.
- **Hand-written.** Please submit your answers to the instructor at the beginning of the class. TA will collect the answer sheets in the first break of the class. **NO LATE SUBMISSION IS ALLOWED**.
- **Programming.** Please upload your source codes in a single .tar or .zip file (e.g., b04901001_hw3.zip) to Ceiba. You must implement your program with **Python 3**.
- **Delay policy.** You have a six-hour delay quota for programming assignments for the whole semester. Once the quota is used up, any late submission will receive no credits.
- **Collaboration policy.** Discussion with other students is strongly encouraged, but you must obtain and write the final solution by yourself. Please specify the references (e.g., the name and student id of your collaborators and/or the Internet URL you consult with) for each **hand-written and programming question** on your answer sheet of the hand-written problems. If you solve some problems by yourself, please also specify "no collaborator". Homeworks without reference specification may not be graded.
- **Academic honesty.** Cheating is not allowed and is against university policy. Plagiarism is a form of cheating. If cheating is discovered, all students involved will receive an F grade for the course (NOT negotiable).

Hand-Written Problems

Note that in this section, you can apply all graph algorithms introduced in class as a black box without further explanation unless specified.

Problem 1 (15%)

Please answer the following questions.

1. (2%) A tic-tac-toe game can be modeled as a directed graph by converting each board position (i.e., a game state) to a vertex and each legal move to an edge. What is the in-degree and out-degree of the board position shown in Figure 1. (It is X's turn.)

   Hint: Any "legal" move is an edge.

2. (3%) Apply Dijkstra's algorithm to the graph shown in Figure 2 starting from vertex S. What is the order of vertices that get removed?

3. (5%) What is the running time of depth-first search, in terms of |V| and |E|, implemented with (1) adjacency list, and (2) adjacency matrix?

4. (5%) An **Euler tour** of a strongly connected, directed graph $G(V, E)$ is a cycle that traverses each edge of G exactly once, although it may visit a vertex more than once. Show that G has an Euler tour **if and only if** in-degree(v) = out-degree(v) for each vertex $v \in V$.

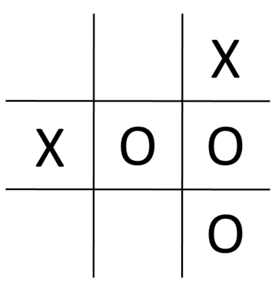

Figure 1: A tic-tac-toe board position

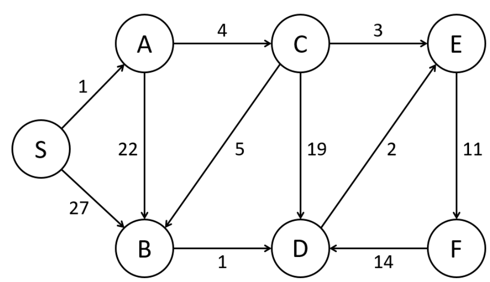

Figure 2: A directed graph

Problem 2 (10%)

A **binary-weighted graph** is a graph where each edge has a weight of either 0 or 1. Given a source vertex S in a binary-weighted graph, please devise a $O(V + E)$ algorithm to find the shortest path from S to all the other vertices. Briefly justify the correctness.

Problem 3 (10%)

Sheldon lives in Pasadena, California. He'd like to join the comic-con that will take place in San Diego in three days. He decides to drive to San Diego; however, he is a poor driver. To avoid a car accident, he labeled each road between Pasadena and San Diego with the probability (a real number in [0, 1]) that he can drive safely on that road. If a car accident happens on his drive, Sheldon will not be able to join the comic-con and will become extremely "irritating"!
Please help his friend Leonard, who lives with Sheldon, to prevent such a tragedy from happening. Describe an efficient algorithm to find a path from Pasadena to San Diego that maximizes the probability of a safe drive by modeling the map as a **directed graph**. Briefly justify the correctness and analyze the running time.

Hint: Model the towns between Pasadena and San Diego as vertices and the road between two towns as two directed edges.

Problem 4 (15%)

Zombie apocalypse suddenly breaks in South Korea! Jonathan and his daughter Amy want to travel from Seoul to Busan to meet with his beloved/her mother Isabelle. They have a map of all train routes between Seoul and Busan, which records the time it will take to travel from one station to the other. However, on some of the train routes, they will be attacked by the zombies. Jonathan has a confidence that he can protect his daughter from the zombie attack for **once** during their trip from Seoul to Busan. However, if they encounter the zombie attack more than once during the trip, they are destined to become one of those horrible monsters. Fortunately, Jonathan also has the information indicating whether each train route is free from zombie attacks or not.
Please help Jonathan design a **polynomial time algorithm** (to the number of train routes and stations) to let them travel from Seoul to Busan **as fast as possible** without becoming a zombie. Briefly justify the correctness and analyze the running time.

Problem 5 (15%)

Given n cities $C_1, C_2, ..., C_n$. The cost of constructing an undirected road between cities C_i and C_j requires cost w_{ij}. The cost of constructing an airport at city C_i is p_i. One can travel between two cities using an airport if both cities have airports. Given all costs w_{ij} and p_i, design a **polynomial time algorithm** to find the minimum cost set of airports and roads such that people from every cities can travel to any other cities. Briefly justify the correctness and analyze the running time.

Programming Problems

Note that in the programming problems, input is from the **standard input** and you should output the result to the **standard output**. You **cannot import any module or library**. Also, you can only implement your program with **Pyhton3**. Only `p1.py` and `p2.py` (if you have finished the bonus problem) should be uploaded. Put the file(s) in a folder named {student id} and compress the folder as .zip or .tar format.

Problem 1 (35%)

Professor Wang traveled to London for an important computer vision conference, ECCV. After receiving his fifth consecutive best paper award in the conference, he realized that no one can beat him in the community of computer vision and it's time to search for something more challenging. Thus, he went to London King's Cross and took a train at Platform $9\frac{3}{4}$ to Hogwarts to start his new life as a wizard.
There are so many magic courses he should take to graduate from Hogwarts. However, he finds that some of the magic courses have prerequisites. For example, before learning `Divination`, he should have learned `Charms` before.

Here comes the question. Given the total number of magic courses to take and a list of prerequisite pairs, return the minimum number of years it takes to graduate from Hogwarts and an ordering of courses he should take to finish all of them. Note that Hogwarts also have two semesters and some of the magic courses are only available in one of the two semesters. Suppose a semester is 0.5 year.

Input Format

The first line contains two integers n and p, indicating the number of courses a student should take to graduate, and the number of prerequisite pairs, respectively. The following n lines contain two integers i and j, where i is the course id (from 0 to $n-1$) and j indicates which semester the course is available (0 for the first semester, 1 for the second semester, and 2 for both of the semesters). The following p lines contain two integers x and y, indicating to take course x, you should have finished course y.

- $1 \leq n \leq 1 \times 10^4$
- $0 \leq p \leq 5 \times 10^5$

Output Format

If there is no legal solution, output a single -1. If there is a legal solution, in the first line, please output how many years it takes to graduate. In the following lines, please output the set of course ids to take in each semester (separated by a space, and -1 for no class in this semester).

Time Limit

15 seconds

Sample Input 1

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

Sample Output 1

```
1.0
1
0
```

Sample Input 2

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

Sample Output 2

```
1.5
1
−1
0
```

Sample Input 3

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

Sample Output 3

```
1.5
0
1 2
3
```

Sample Input 4

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

Sample Output 4

```
−1
```

Problem 2 (bonus 20%)

We have calculated how many years it takes for Prof. Wang to graduate from Hogwarts. However, we forget to take the "magic power" a person possesses into consideration. Taking a magic course requires several points of "magic power". **In each semester**, the sum of "magic power" requirement of every courses cannot exceed a person's magic power limit; otherwise, he will become a Muggle.

Given the magic power limit m of Prof. Wang, please determine how many years it takes for him to graduate from Hogwarts and an ordering of courses he should take.

Input Format

The first line contains three integers n, p and m, where n and p are the same as in problem 1, and m indicates the magic power limit of Prof. Wang. The following n lines contain three integers i, j, and k, where i is the class id (from 0 to $n-1$), j indicates which semester the class is available (0 for the first semester, 1 for the second semester, and 2 for both of the semesters), and k indicates the magic power requirement for this course. The following p lines contain two integers x and y, indicating to take course x, you should have finished course y.

- $1 \leq n \leq 1 \times 10^5$
- $0 \leq p \leq 5 \times 10^6$
- $1 \leq m \leq 2 \times 10^3$

Output Format

There is always a legal solution for each test case. In the first line, please output how many years it takes to graduate. In the following lines, please output the set of class ids to take in each semester (separated by a space, and -1 for no class in this semester).

Time Limit

30 seconds

Grading Policy

- Output any legal solution correctly in the time limit: bonus 10 points
- Help Prof. Wang graduate earlier than 50% of solutions in our class: bonus 5 points
- Help Prof. Wang graduate earlier than 80% of solutions in our class: bonus 5 points

Sample Input 1

3 2 25
0 2 25
1 2 1
2 2 1
2 0
2 1

Sample Output 1

1.5
0
1
2