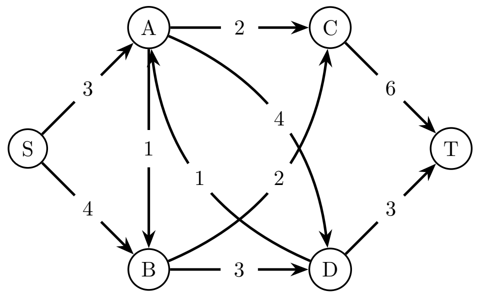
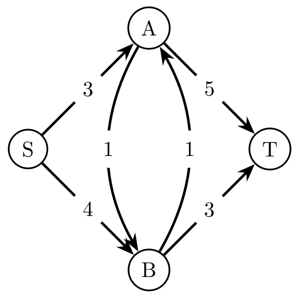
CSCI-570 Homework 4

March 25, 2020

1 Compute Max-Flow And Min-Cut (10 points)

Compute the max-flow and list all the minimum s-t cuts in the flow network of the following graphs. For the max-flow, you need to give a feasible flow that achieves that maximum value.





2 Escape From the Building (15 points)

In this problem, we need to decide whether there is a feasible plan for all the persons in a building to escape when they meet some emergency issues. More specifically, a building is described as an n by n grid and the position of p persons are represented as the integer points $(x_1, y_1), \dots, (x_p, y_p)$ in the building. Note that to ensure safety, we don't allow any intersection between the paths of any two person. Therefore, your task is to decide whether there exist p vertex-disjoint paths from their starting points to any p different points on the boundary of the grid. Give an algorithm polynomial in n and prove the correctness of it.

3 Install Software to Your New Computer (15 points)

Suppose that you have just bought a new computer and you want to install software on that. Specifically, two companies, which you can think of like Microsoft and Apple, are trying to sell their own copy of n different products, like Operation System. Spread Sheet, Web Browser. For each product i, $i \in \{1, 2, \ldots, n\}$, we have

- the price $p_i \geq 0$ that Microsoft charges and the price $p'_i \geq 0$ that Apple charges.
- the quality $q_i \geq 0$ of Microsoft version and the quality $q'_i \geq 0$ of Apple version.

For example, Apple may provide a better Web Browser Safari, but Microsoft a better Word Processor. You want to assemble your favorite computer by installing exactly one copy of each of the n products, e.g. you want to buy one operating system, one Web Browser, one Word Processor, etc. However, you don't want to spend too much money on that. Therefore, your goal is to maximize the quality minus total price.
However, as you may know, the products of different companies may not be compatible. More concretely, for each product pair (i, j), we will suffer a penalty $\tau_{ij} \geq 0$ if we install product i of Microsoft and product j of Apple. Note that τ_{ij} may not be equal to τ_{ji} just because Apple's Safari does not work well on Microsoft Windows doesn't mean that Microsoft's Edge does not work well in Mac-OS. We assume that products are always compatible internally, which means that there is no penalty for installing two products from the same company. All pairwise penalties will be subtracted from the total quality of the system.

Your task is then to give a polynomial-time algorithm for computing which product i to purchase from which of the two companies (Apple and Microsoft) for all $i \in \{1, 2, \ldots, n\}$, to maximize the total system quality (including the penalties) minus the total price. Prove the correctness of your algorithm. (Hint: You may model this problem as a max-flow/min-cut problem by constructing your graph appropriately.)

4 Jumping Frogs (15 points)

Somewhere near the Algorithmville, a number of frogs are standing on a number of lotus leaves. As they are social animals (and yes, they are never infected by coronavirus!), the frogs would like to gather together, all on the same lotus leaf. The frogs do not want to get wet, so they have to use their limited jump distance d to get together by jumping from piece to piece. However, these lotus leaves just started to grow, they will get damaged further by the force needed to jump to another leaf. Fortunately, the frogs are real experts on leaves, and know exactly how many times a frog can jump off each leaf before it sinks and become unavailable. Landing on leaves does not damage it. You have to help the frogs find a leaf where they can meet.

In this question, we will get the position of N lotus leaves. For each $i \in [N]$, we know its position (x_i, y_i), the number of frogs n_i on that leaf and the number of jumps m_i before it sinks. The distance between two leaves (x_i, y_i) and (x_j, y_j) is defined as $|x_i - x_j| + |y_i - y_j|$. Design a polynomial algorithm to determine whether whether each lotus leaf can hold all frogs for a party. The output is an array with length N, containing yes/no solution. Prove the correctness of your algorithm.

5 Preparing for the Exams (15 points)

My friend Leo wants to have a emergency plan for his final exams on University of Southern Algorithmville. He has N subjects to prepare for, and for each subject, his score is determined only by the time he spend on learning. It's not surprising that Leo found out he actually spent **zero** time on preparing before.

At least he knows when he can start learning all of these subjects. For each subject i there is a start time s_i when he can get all materials ready to start learning. And there is also a ending time e_i for each subject, when his learning materials expire and he can't learn anymore. We know that s_i and e_i are integers, and Leo can only dedicate to a single subject within each time phase.

In Universtiy of Southern Algorithmville (USA), a student's total grade is the minimum grade among all subjects. Leo wants you to help him find out the best outcome. Given N subjects and their time intervals (s_i, e_i), design an algorithm to find out the maximum time possible for the least prepared subject. Prove the correctness of your algorithm. (Hint: It's not enough to use the network flow algorithm alone to determine the answer.)

6 Help Kumiko! (15 points)

Kumiko is a member of a high school music group and she is in charge of counting the monthly community fee for each members. This work should be easy but the members are trying to make Kumiko's life harder. They never pay the fee in an integer amount! The following form is what Kumiko has recorded.

	Jan	Feb	Mar	Apr	Total Sum
Reina	13.12	17.04	28.92	16.92	76
Mizore	18.80	20.98	22.95	13.27	76
Shuuichi	0.08	0.98	0.13	107.81	109
Total Sum	32	39	52	138	261

Fortunately, Kumiko finds that the total sum of each month and of each member are integers. To make the table cleaner, she wants to round all data in the table into integers without changing any column or row sum. Therefore, no member is paying more in the table and the amount of money in each month keeps the same! Specifically, each fractional number can be rounded either up or down. For example, a good rounding for the data above would be as follows.

	Jan	Feb	Mar	Apr	Total Sum
Reina	13	17	29	17	76
Mizore	19	21	23	13	76
Shuuichi	0	1	0	108	109
Total Sum	32	39	52	138	261

As you have just learned network flow and related algorithm, you want to tell Kumiko that it is an easy job and she can ALWAYS do the above process. More specifically, consider there are m members, n months and a data matrix $\{M_{ij}\}_{(i,j)=(1,1)}^{(m,n)}$. The sum of each row and column is integer. You want to round each entry M_{ij} to either $\lceil M_{ij} \rceil$ or $\lfloor M_{ij} \rfloor$ without changing the sum of each row and column. Give an polynomial time algorithm based on network flow to obtain such rounding and show the correctness of it. (Hint: (1). You may start with considering the case when $M_{ij} \in [0, 1]$ and then generalize it. (2). You may notice the fact that the network flow algorithm you have learned always outputs an "integral" max-flow.)

7 Edges that Increase Max-Flow (15 points)

Given a graph $G = (V, E)$, the source-sink pair (s, t) and capacity of edges $\{c_e \geq 0 \mid e \in E\}$, design a polynomial-time algorithm to find a set of edges S, such that for every edge $e \in S$, increasing c_e will lead to an increase of max-flow value between s and t. Show the correctness of your algorithm.