

Homework 9

Deadline: Monday (25 March 2024) 11:59 PM

- 1) There is a precious diamond that is on display in a museum at m disjoint time intervals. There are n security guards who can be deployed to protect the precious diamond. Each guard has a list of intervals for which he or she is available to be deployed. Each guard can be deployed to at most M time slots and has to be deployed to at least L time slots. Design an algorithm that decides if there is a deployment of guards to intervals such that each interval has either one or two guards deployed. (10 pts)
- 2) Consider LAX, a notoriously busy airport with many arriving passengers who want to get to their destinations as soon as possible. There is an available fleet of n Uber drivers to accommodate all passengers. However, there is a traffic regulation at the airport that limits the total number of Uber drivers at any given hour-long interval to $0 \leq k < n$ simultaneous drivers. Assume that there are p time intervals. Each driver provides a subset of the time intervals he or she can work at the airport, with the minimum requirement of a_j hour(s) per day and the maximum b_j hour(s) per day. Lastly, the total number of Uber drivers available per day must be at least m to maintain a minimum customer satisfaction and loyalty. Design an algorithm to determine if there is a valid way to schedule the Uber drivers with respect to these constraints. (20 pts)
- 3) Counter Espionage Academy instructors have designed the following problem to see how well trainees can detect SPY's in an $n \times n$ grid of letters S, P, and Y. Trainees are instructed to detect as many disjoint copies of the word SPY as possible in the given grid. To form the word SPY in the grid they can start at any S, move to a neighboring P, then move to a neighboring Y. (They can move north, east, south or west to get to a neighbor.) The following figure shows one such problem on the left, along with two possible optimal solutions with three SPY's each on the right. Give an efficient network flow-based algorithm to find the largest number of SPY's. (20 pts)

Note: We are only looking for the largest **number** of SPYs not the actual location of the words. No proof is necessary.

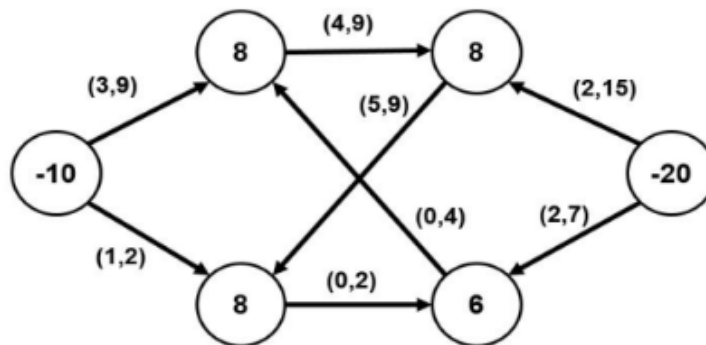
Y	S	S	P
P	S	P	Y
S	S	Y	P
Y	S	P	S

Y	S	S	P
P	S	P	Y
S	S	Y	P
Y	S	P	S

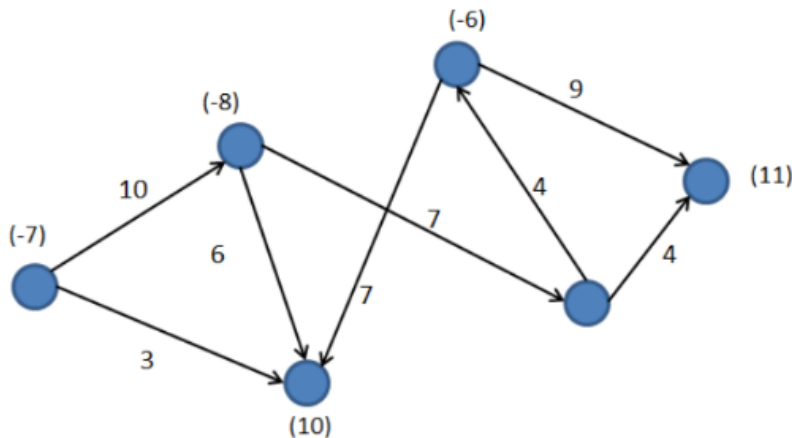
Y	S	S	P
P	S	P	Y
S	S	Y	P
Y	S	P	S

UNGRADED

- 4) In the network below, the demand values are shown on vertices. Lower bounds on flow and edge capacities are shown as (lower bound, capacity) for each edge. Determine if a feasible circulation exists or not. If it does, show the feasible circulation. If it does not, show the cut in the network that is the bottleneck. Show all your steps.



- 5) The following graph G is an instance of a circulation problem with demands. The edge weights represent capacities and the node weights (in parentheses) represent demands. A negative demand implies source.



- (i) Transform this graph into an instance of max-flow problem.
- (ii) Now, assume that each edge of G has a constraint of lower bound of 1 unit,

i.e., one unit must flow along all edges. Find the new instance of max-flow problem that includes the lower bound constraint.

- 6) Determine if it is possible for flow to circulate within the given network depicted as graph G . The demand values, indicated on vertices, represent either supply (if positive) or demand (if negative). Each edge also has specified lower bounds on flow and capacity. Demonstrate all necessary steps to ascertain if a feasible circulation exists within this graph. (20 pts)
- (a) Reduce the Feasible Circulation with Lower Bounds problem to a Feasible Circulation problem without lower bounds.
 - (b) Reduce the Feasible Circulation problem obtained in part (a) to a Max Flow problem in a Flow Network.
 - (c) Solve the resulting Max Flow problem and explain whether there is a Feasible Circulation in the original G .
- 7) USC Admissions Center needs your help in planning paths for Campus tours given to prospective students or interested groups. Let USC campus be modeled as a weighted, directed graph G containing locations V connected by one-way roads E . On a busy day, let k be the number of campus tours that have to be done at the same time. It is required that the paths of campus tours do not use the same roads. Let the tour have k starting locations $A = \{a_1, a_2, \dots, a_k\} \subseteq V$. From the starting locations, the groups are taken by a guide on a path through G to some ending location in $B = \{b_1, b_2, \dots, b_k\} \subseteq V$. Your goal is to find a path for each group i from the starting location, a_i , to any ending location b_j such that no two paths share any edges, and no two groups end in the same location b_j .
- (a) Design an algorithm to find k paths $a_i \rightarrow b_j$ that start and end at different vertices, such that they do not share any edges.
 - (b) Modify your algorithm to find k paths $a_i \rightarrow b_j$ that start and end in different locations, such that they do not share any edges or vertices.