Homework 9

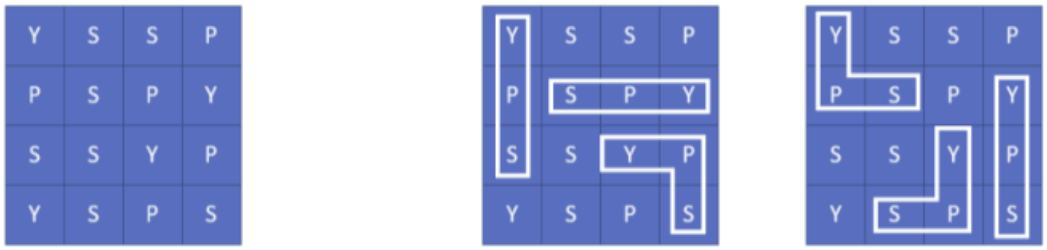
1. **There is a precious diamond that is on display in a museum at m disjoint time intervals. There are n security guards who can be deployed to protect the precious diamond. Each guard has a list of intervals for which he or she is available to be deployed. Each guard can be deployed to at most M time 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)**
   Ans:
   Algorithm:
   Create a flow network with nodes representing intervals, guards, a source node, and a sink node.
   Connect the source node to each interval node with an edge of capacity 2, representing the requirement that each interval must have either one or two guards.
   Connect each guard node to the sink node with an edge of capacity M, representing the maximum number of times slots a guard can be deployed.
   Connect each guard node to the intervals it is available for deployment with edges of infinite capacity.
   Connect overlapping intervals with edges of infinite capacity to ensure that no guard is assigned to overlapping intervals simultaneously.

   Now, find the maximum circular flow using the Ford-Fulkerson algorithm. If the maximum circular flow saturates all edges going out from the source node (each interval has either one or two guards deployed), then the problem is feasible.

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 <= 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 aj hour(s) per day and the maximum bj 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)**
   Ans:
   Algorithm:
   Create a flow network with nodes representing time intervals, drivers, a source node, and a sink node.
   Connect the source node to each driver node with edges of capacity bj - aj, representing the range of hours each driver can work.

Connect each driver node to the time interval nodes representing the hours they are available with edges of infinite capacity.
Connect the time interval nodes to the sink node with edges of capacity 1, representing the requirement of having at least one driver available for each hour.
Connect overlapping time intervals with edges of infinite capacity to ensure drivers are not scheduled for overlapping hours.
Now, find the maximum circular flow using the Ford-Fulkerson algorithm. If the maximum circular flow saturates all edges going out from the source node (indicating that all drivers' available hours are utilized) and satisfies the minimum total number of drivers required (m), then the problem is feasible.

3. **Counter Espionage Academy instructors have designed the following problem to see how well trainees can detect SPY's in an nXn 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.



Ans:
Algorithm:
Create a flow network with source node 's' and a sink node 't'. Create a node for each cell (i, j) in the grid.
For each cell (i, j) in the grid:
If the cell contains 'S', add an edge from the source s to (i, j) with capacity 1.
If the cell contains 'Y', add an edge from (i, j) to the sink t with capacity 1.
If the cell contains 'P', add edges from (i, j) to its neighboring cells in the grid with capacity 1.
Next, use Ford-Fulkerson algorithm to find the maximum flow in the network.
Return the maximum flow that represents the maximum number of disjoint copies of the word SPY that can be formed in the grid.