

Pre-tutorial questions

Do you know the basic concepts of this week's lecture content? These questions are only to test yourself. They will not be explicitly discussed in the tutorial, and no solutions will be given to them.

1. Bipartite matching.
 - (a) What is a bipartite matching?
 - (b) How can one use flow networks to find a maximum bipartite matching?
 - (c) What does the marriage theorem state?
 2. Edge disjoint paths
 - (a) What does it mean that two paths in a graph are edge disjoint?
 - (b) How can one use flow networks to count the maximum number of edge disjoint paths?
 3. Extensions of flow networks
 - (a) What is flow networks with node supplies and demands?
 - (b) What is a circulation in a network with supplies and demands?
 - (c) How can one use flow networks to model supply and demand?
-

Tutorial

Problem 1

Suppose you're in charge of managing a fleet of airplanes and you'd like to create a flight schedule for them. Here's a very simple model for this. Your market research has identified a set of m particular flight segments that that would be very lucrative if you could serve them; flight segment j is specified by four parameters: its origin airport, its destination airport, its departure time, and its arrival time. An example of six flights segments you'd like to serve with your planes over the course of a single day may include:

1. Brisbane (depart 6 am) - Sydney (arrive 7 am)
2. Sydney (depart 6:30 am) - Melbourne (arrive 8 am)
3. Sydney (depart 8 am) - Cairns (arrive 11 am)
4. Melbourne (depart 12 am) - Adelaide (arrive 2 pm)
5. Townsville (depart 2:15 pm) - Brisbane (arrive 4 pm)
6. Adelaide (depart 4:00 pm) - Perth (arrive 7 pm)

It is possible to use a single plane for a flight segment i , and then later for flight segment j , provided that:

1. the destination of i is the same as the origin of j and there's enough time to perform maintenance on the plane in between; or

2. you can add a flight segment in between that gets the plane from the destination of i to the origin of j with adequate time in between.

For the example above you can group flights 1, 3, and 5 together by allocating an hour maintenance between each flight and adding an additional 1 hour flight between Cairns and Townsville for example.

Your goal based on the two conditions stated above is to determine if it is possible to service all m flights using at most k planes. In order to do this you will need to reuse planes as efficiently as possible.

Problem 2

Given an **undirected** graph $G = (V, E)$ and edge capacities $c : E \rightarrow \mathbb{R}_+$ and two vertices s and t , design an algorithm for computing a minimum capacity s - t cut.

Problem 3

We define a new version of the game Capture the Flag. The game is played in a maze that consists of a large number of rooms that are connected by doors. Each door connects two rooms. Each player starts in a given room in the maze (different players may start in different rooms). From each room a player can decide to move to any adjacent room, going through a door, as long as the door is open. All doors are open in the beginning of the game, however, whenever a player crosses a door, the door will be closed and locked. That means, no other player will be able to go through that door, if they happen to find themselves in the same room at some point. So, when a player arrives in a new room, they can choose to go through any open door, but that door will be closed, and not available to anyone else from that point on. Some rooms contain a flag, and all flags are the same. Each player is trying to find a flag (any flag will do). A player can only pick up one flag. All players belong to the same team therefore they want to collect all flags if possible.

Here is the problem you need to solve. You are given the complete description of the maze, which consists of a set of rooms R , and a complete description of the set of doors D (which are pairs of rooms that are connected by a door). You are also given a set $S \subset R$, the initial k locations (rooms) of your team of k players. And you are given the set $F \subset R$, which are the k rooms that contain a flag. Your task is to check whether it is possible to find paths that your k players can follow in order to find all k flags, and no two paths cross the same door. Note that a given maze may not allow this, since doors can only be used once.

Given R, D, S and F , describe how to decide whether such a maze allows the collection of all flags or not.

Problem 4

Let G be a flow network such that, for every edge e in the network, $c(e)$ is an even integer.

1. Argue that the maximum value of a flow is an even integer.
2. Show that there is a maximum flow f such that, for every edge e , the flow over e is an even integer.

Problem 5

A d -regular graph is one where every vertex has degree d . Prove that every d -regular bipartite graph has a perfect matching.

Problem 6

In class we argued that Ford-Fulkerson runs for at most C iterations. If C is too large, this may take too long. Luckily, a variant of the algorithm can be shown to run for at most $O(m \log C)$ iterations. The idea is very simple: Select an augmenting path maximizing the minimum residual capacity along the path. Show how to modify Dijkstra's algorithm to find such a path in the residual graph G^f .