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. Flow network $G = (V, E)$.
   (a) Is G directed or undirected?
   (b) What does it mean that an edge has a capacity?
   (c) What is a flow f in G?
   (d) Why is the flow of an edge bounded by the capacity of an edge?
   (e) What are the capacity and conservation constraints?
2. Min Cut (A, B) of G.
   (a) What is a cut (A, B) of G?
   (b) What is the capacity of a cut?
   (c) What is the flow of a cut?
3. Min cut vs. max flow
   (a) Why is the value of a cut an upper bound on the maximum flow?
   (b) What does the Min Cut - Max Flow theorem state?
4. Ford-Fulkerson
   (a) The Ford-Fulkerson algorithm iteratively increases the flow. How is this done in each iteration?
   (b) Can you upper bound the number of iterations performed by the Ford-Fulkerson algorithm?

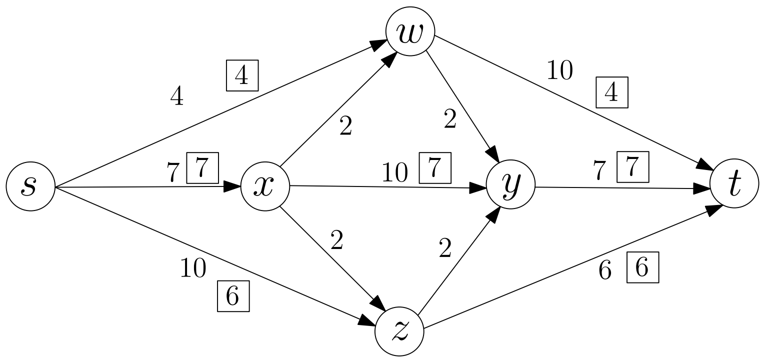
Tutorial

Problem 1
Let $G = (V, E)$ be an arbitrary flow network, with source s, sink t, and edge capacities $c : E \to \mathbb{Z}^+$. Decide whether each of the following statements are true of false. If it is true, give a short explanation. If it is false, give a counterexample.

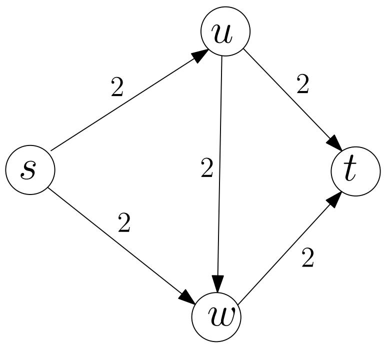
1. If f is a maximum s-t flow in G, then f saturates every edge out of s; that is, $f(e) = c(e)$ for each edge e coming out of s.
2. Let (A, B) be a minimum s-t cut with respect to c. Define new capacities $\hat{c}(e) = c(e) + 1$ for each $e \in E$. Then (A, B) is a minimum s-t cut with respect $\hat{c}$.

Problem 2
The figure below shows a flow network on which an $s - t$ flow is shown. The capacity of each edge appears as a label next to the edge, and the numbers in boxes give the amount of flow sent on each edge. (Edges without boxed numbers have no flow being sent on them.)

1. What is the value of this flow?
2. Is this a maximum $s-t$ flow in this graph? If not, find a maximum $s-t$ flow.
3. Find a minimum $s-t$ cut. (Specify which vertices belong to the sets of the cut.)



Problem 3
Find all minimum $s-t$ cuts in the following graph. The capacity of each edge appears as a label next to the edge.



Problem 4
Design a linear time algorithm that given a flow f verifies that f is maximum. If the flow is maximum your algorithm should output "yes", otherwise, it should output "no". No other action is required.

Problem 5
Consider a generalization of the maximum flow problem where vertices have capacities. An instance is defined by a directed graph $G = (V, E)$, a pair of vertices s and t, and vertex capacities $c : V \to Z^+$. A flow $f : E \to \mathbb{Z}^+$ is feasible if $f^{out}(u) \leq c(u)$. The objective is to find a maximum feasible s-t flow.
    Design an efficient algorithm for this problem.

Problem 6
Consider a generalization of the minimum cut problem where we want to separate two sets of vertices. An instance is defined by a graph $G = (V, E)$, a pair of disjoint subsets of vertices $S, T \subseteq V$, and edge capacities $c : E \to \mathbb{Z}^+$. The objective is to find a minimum capacity cut (A, B) such that $S \subseteq A$ and $T \subseteq B$.
    Design an efficient algorithm for this problem.

Problem 7
We want to wirelessly connect a set of mobile computers to a set of stationary base stations. Each computer u has an effective transmission radius r_u. Suppose we know the spatial location of each computer and each base station. Our goal is to assign each computer to some station within its transmission radius. Define the load of a station to be the number of computers assigned to it.

Your task is to design a polynomial time algorithm that will produce a feasible computer-station assignment minimizing the maximum station load.

Problem 8 - for ADVANCED
In last weeks advanced lecture we discussed sequence alignment. In the version we considered we assumed that the penalty for a gap was δ. However, in practice the penalty for a gap is not a linear function on the length of the gap, usually it it's increases much slower. For this question assume the gap penalty is $gap(k)$ for $1 \leq k \leq n$. Formulate a dynamic program for this generalised version of the problem.