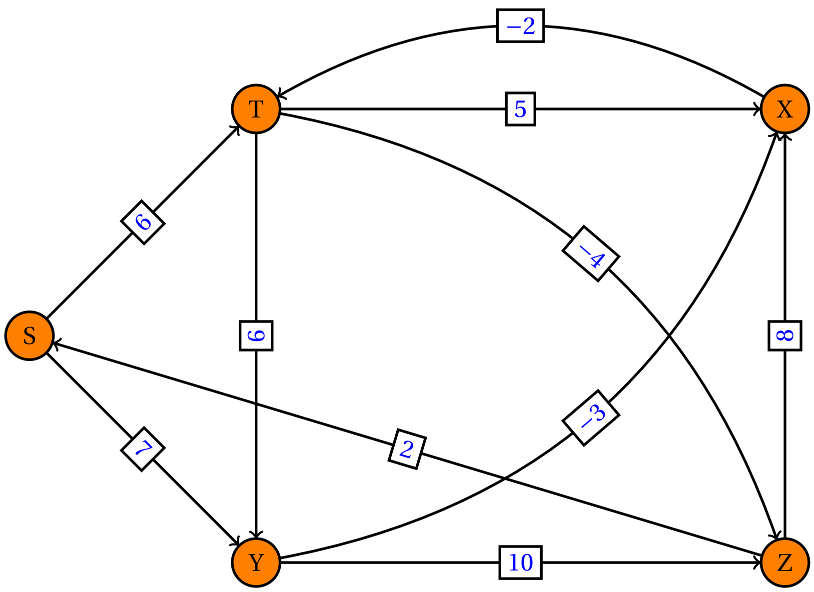
Problem session 5

2nd October 2019

Exercise 1. *Weight: 10%*

Run the Bellman-Ford algorithm on the following directed graph, using vertex s as the source. In each pass, relax edges in the the following order: (t,x), (t,y), (t,z), (x,t), (y,x), (y,z), (z,x), (z,s), (s,t), (s,y). Show the d and π values after each pass.



Exercise 2. *Weight: 25%*

Shortest path algorithms can be applied in currency trading. Here, we want to exchange money from one currency to another. The problem is that not all conversions are possible: given two currencies A and B, one can exchange money in currency A to get B, but the reverse is not always true. Let us consider an exchange graph $G = (V, E)$ between currencies, giving all possible transactions. The graph is directed (for instance, we might be able to exchange from A to B but not from B to A). The exchange function is given by function c such that:

- an amount S in currency A costs $S \cdot c(A, B)$ in currency B
- $c(A, B)$ is defined iff (A, B) is an edge of G
- $c(A, B) > 0$

The exchange graph $G' = (V, E, c)$ is weighted by the exchange function. Let $a_1, a_2, \ldots, a_n$ be various currencies (for instance a_1 might be dollars, a_2 pounds, and a_3 euros). An exchange sequence $a_1, a_2, \ldots, a_k$ is the conversion of money in currency a_1 to a_k through intermediate currencies $a_2 \ldots a_{k-1}$

1. What is the exchange rate from a_1 to a_k in the exchange sequence $a_1, a_2, \ldots, a_k$?
2. Under what condition (consider here G') can someone become infinitely rich by exchanging money?
3. Given 2 exchange sequences S_1 and S_2, the best one is the one with the highest rate. Suppose now that we know a sequence S_1 to exchange currency A in currency B, and S_2 to exchange currency A in currency C. Suppose now that there exists an edge (C, B) in G. Write the relaxation condition comparing the sequence rates S_1, and S_2 then (C, B). Keep the best one.
4. Write a modified version of the Bellman-Ford algorithm, using this modified relaxation condition, and give the best exchange rates from currency A to all others. Why is that correct?

Exercise 3. *Weight: 20%*

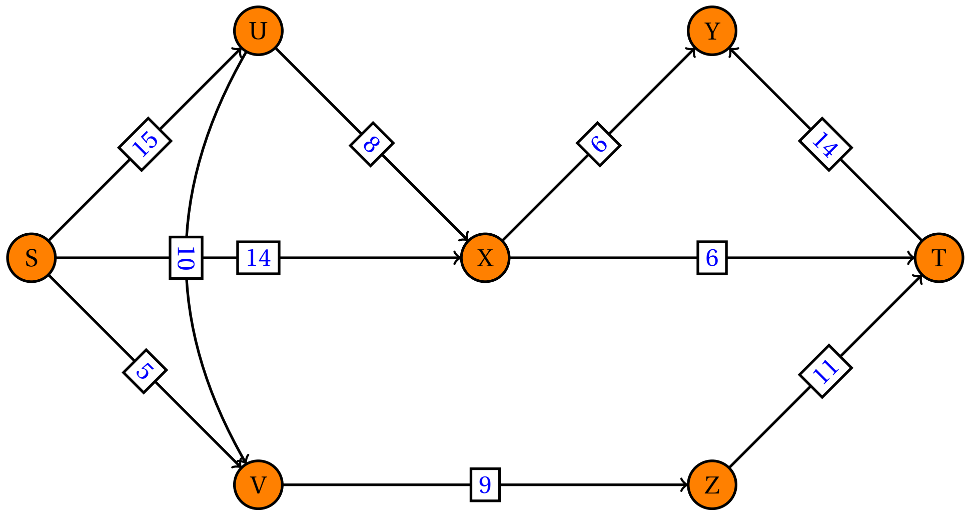
You are given a directed graph $G = (V, E)$ with (possibly negative) weighted edges, along with a specific node $s \in V$ and a tree $T = (V, E'), E \subseteq E$. Give an algorithm that checks whether T is a shortest-path tree for G with starting point s. Your algorithm should run in linear time.

Exercise 4. *Weight: 10%*

A student proposes the following algorithm for finding the shortest path from node s to node t in a directed graph with some negative edges: add a large constant to each edge weight so that all the weights become positive, then run Dijkstra's algorithm starting at node s, and return the shortest path found to node t. Is this a valid method? Either prove that it works correctly, or give a counterexample.

Exercise 5. *Weight: 10%*

Run the Ford-Fulkerson algorithm on the following flow network. For each iteration, pick the shortest augmenting path from S to T and give the residual network. Finally, give the value of the maximum flow and a minimum cut (S, T).



Exercise 6 . *Weight: 25%*

A server S is connected to a computer T via a network composed of nodes A, B, C and D. The transfer capacities (in Mbit/s) are the following:

	A	B	C	D	T
S	2	6	1		
A		3		7	
B				3	5
C		2		6	
D	3				4

The user of machine T downloads a very big file from server S. We want to find the routing that maximizes the bit rate.

1. Which algorithmic problem does this problem corresponds to? Which algorithm may we use?

2. Run the algorithm.

3. What is the maximal bit rate? Which routing enables it?