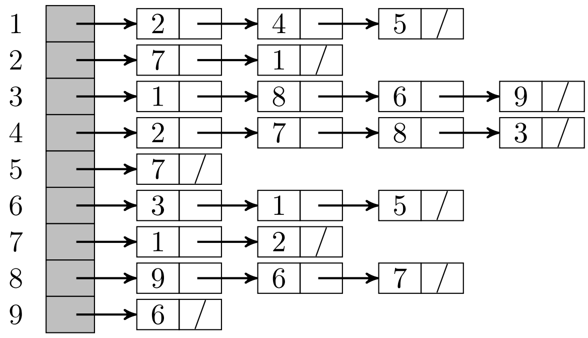
Exercise VIII, Algorithms I 2024-2025

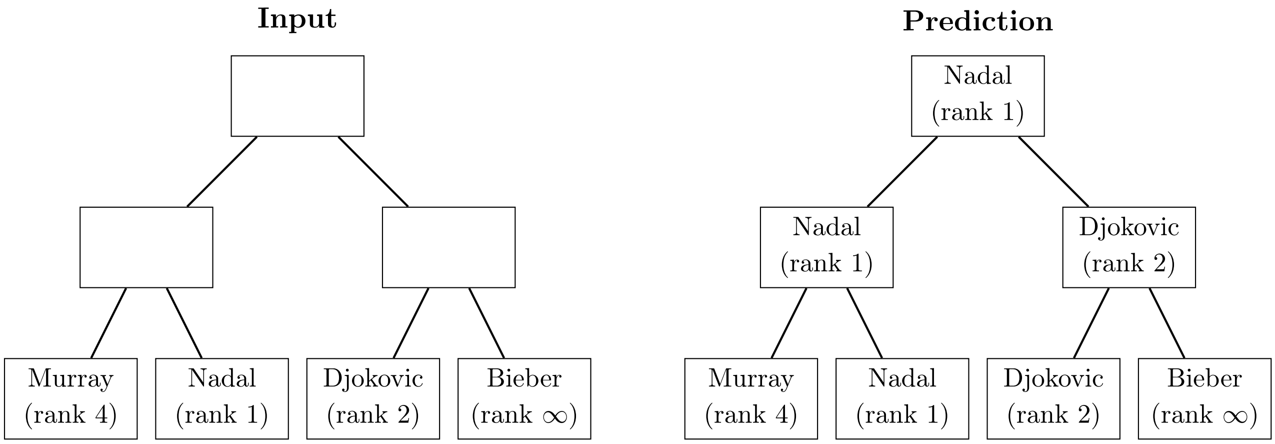
These exercises are for your own benefit. Feel free to collaborate and share your answers with other students. There are many problems on this set, solve as many as you can and ask for help if you get stuck for too long. Problems marked * are more difficult but also more fun :).

Basic Graph Algorithms

1 What are the numbers of edges going into and out of vertex 7 in the graph represented by the following adjacency-list?



2 (Exercise 22.4-3) Give an algorithm that determines whether or not a given undirected graph $G = (V, E)$ contains a cycle. Your algorithm should run in $O(|V|)$ time, independent of $|E|$.



3 *(old exam question)* **Australian Open.** The draw of Australian Open was recently announced. In tennis each match is between two players and the winner progresses to the next round. This naturally leads to a complete binary tree structure of the tournament. At the leaves, we have all the players in the tournament. At the next level, we have those that won their first match, and so on. In particularly, the root of the tree contains the winner of the tournament. We are interested in predicting the outcome of *every match* in Australian Open 2014. To do this we use the following simplifying assumption: a better ranked player always wins over a player with worse rank.

Consider the figure below for an example. We have four players entering the tournament of various rankings. The prediction tree then predicts the winner of each match. For example, as Rafael Nadal is currently ranked number one and Andy Murray is ranked number four, Rafael Nadal wins against Andy Murray. Similarly, we predict that Nadal wins against Djokovic in the final.

Design and analyze an efficient algorithm for the Australian Open prediction problem:

Input: The root of a complete binary tree (the draw) with n players as leaves. Each player/node has a *name* and a *ranking* that is initially empty for nodes that are not leaves. In addition, each node has pointers to its *left child*, its *right child* and *its parent*. Finally, no two players have the same ranking.

Output: A complete binary tree (the prediction) where each node contains the player (his name and rank) that has reached this stage assuming that better ranked players always win over worse ranked players.

Your algorithm should run in *linear time* in the number of players.

4 (*, Exercise 22.2-7) There are two types of professional wrestlers: “babyfaces” (“good guys”) and “heels” (“bad guys”). Between any pair of professional wrestlers, there may or may not be a rivalry. Suppose we have n professional wrestlers and we have a list of r pairs of wrestlers for which there are rivalries. Give an $O(n+r)$-time algorithm that determines whether it is possible to designate some of the wrestlers as babyfaces and the remainder as heels such that each rivalry is between a babyface and a heel. If it is possible to perform such a designation, your algorithm should produce it.

5 (Exercise 22.5-1) How can the number of strongly connected components of a graph change if a new edge is added?

6 (Exercise 22.5-3) Professor Bacon claims that the algorithm for strongly connected components would be simpler if it used the original (instead of the transpose) graph in the second depth-first search and scanned the vertices in order of *increasing* finishing times. Does this simpler algorithm always produce correct results?

Flow Networks

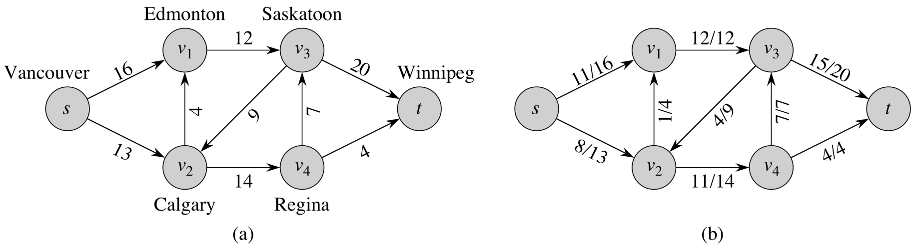


Figure 1. Example of a flow network (on the left) and of a flow (on the right).

7 (Exercise 26.2-2) In Figure 1(b) what is the flow across the cut $(\{s, v_2, v_4\}, \{v_1, v_3, t\})$? What is the capacity of this cut?