Note: Your TA may not get to all the problems. This is totally fine, the discussion worksheets are not designed to be finished in an hour. The discussion worksheet is also a resource you can use to practice, reinforce, and build upon concepts discussed in lecture, readings, and the homework.

1 (MT2 Practice) MST True/False

For each of the following, state if it's true or false. If true, justify your answer, if false give a counterexample.

In each problem, assume G is a weighted undirected graph with non-negative edge weights. You may use the following fact in your justifications without proof: For any MST T of a graph, there is some sorted ordering of the edges such that Kruskal's will output T if it considers adding edges in this order.

(a) G has a unique minimum spanning tree if and only if all edges in G have distinct edge weights.

(b) Suppose all edges in G have weight at most W, and let G' have the same vertices and edges as G, but edge e has weight $W - w_e$ in G' instead of w_e. Then the maximum weight spanning tree of G is the minimum spanning tree of G'.

(c) Let G' have the same vertices and edges as G, but edge e has weight w_e^2 in G' instead of w_e. Then any minimum spanning tree in G' is also a minimum spanning tree in G.

(d) If the smallest edge weight is 1, every MST has the same number of edges with weight 1.

2 (MT2 Practice) Huffman and LP

Consider the following Huffman code for characters a, b, c, d: $a = 0, b = 10, c = 110, d = 111$.

Let f_a, f_b, f_c, f_d denote the fraction of characters in a file (only containing these characters) that are a, b, c, d respectively. Write a linear program with variables f_a, f_b, f_c, f_d to solve the following problem: What values of f_a, f_b, f_c, f_d that can generate this Huffman code result in the Huffman code using the most bits per character?

3 (MT2 Practice) Discrete Golf

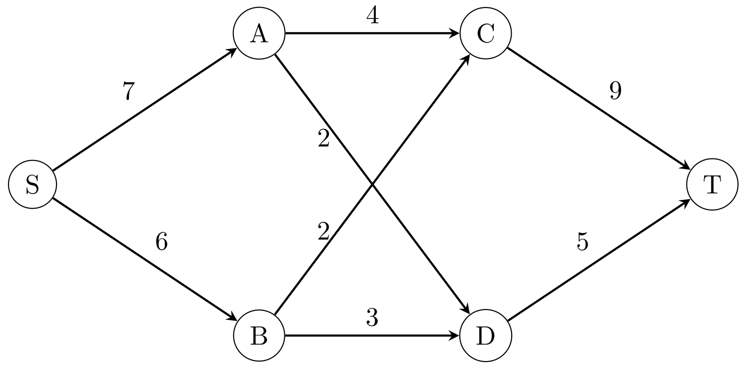
Professor Vazirani loves to play golf, specifically a version called discrete golf. The goal of discrete golf is to hit a golf ball from checkpoint 1 to checkpoint n on a golf course in as few strokes as possible. Based on the terrain of the course, he knows that from checkpoint i, he can hit the ball up to $d(i) \geq 1$ checkpoints away in one stroke. That is, if the ball is at checkpoint i, he can hit the ball to any of checkpoints $i + 1, i + 2 \dots i + d(i)$ in one stroke.

(a) Suppose he hits the ball as far as possible every stroke, i.e. if he is at checkpoint i, he hits the ball to checkpoint $i + d(i)$. Give a counterexample where this greedy algorithm does not achieve the minimum number of strokes needed to reach checkpoint n from checkpoint 1.

(b) Suppose that all $d(i)$ are at most D. Give a $O(nD)$ time algorithm for computing the minimum number of strokes Professor Vazirani needs to reach point n. How much memory does this algorithm need?

4 (MT2 Practice) Bottleneck Edges

Consider the following network (the numbers are edge capacities):



(a) Find the following:

- A maximum flow f, specified as a list of $s - t$ paths and the amount of flow being pushed through each.
- A minimum cut (the set of edges with the smallest total capacity, whose removal disconnects S and T), specified as a list of edges that are part of the cut.

(b) Draw the residual graph G_f (along with its edge capacities). In this residual network, mark the vertices reachable from S and the vertices from which T is reachable.

(c) An edge of a network is called a *bottleneck edge* if increasing its capacity results in an increase in the maximum flow. List all bottleneck edges in the above network.

(d) Give a very simple example (containing at most four nodes) of a network which has no bottleneck edges.

(e) Give an efficient algorithm to identify all bottleneck edges in a network. (Hint: Start by running the usual network flow algorithm, and then examine the residual graph.)

5 (MT2 Practice) Permutation Games

A permutation game is a special form of zero-sum game. In a permutation game, the payoff matrix is n-by-n, and has the following property: Every row and column contains exactly the entries $p_1, p_2, \dots p_n$ in some order. For example, the payoff matrix might look like:

$$P = \begin{bmatrix} p_1 & p_2 & p_3 \\ p_2 & p_3 & p_1 \\ p_3 & p_1 & p_2 \end{bmatrix}$$

Given an arbitrary permutation game, describe the row and column players' optimal strategies, justify why these are the optimal strategies, and state the row player's expected payoff (that is, the expected value of the entry chosen by the row and column player).