Note: Your TA probably will not cover all the problems. This is totally fine, the discussion worksheets are not designed to be finished in an hour. They are deliberately made long so they can serve as a resource you can use to practice, reinforce, and build upon concepts discussed in lecture, readings, and the homework.

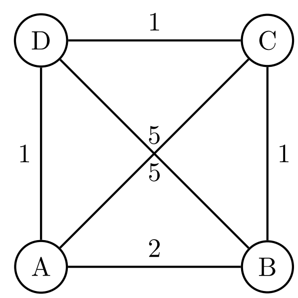
1 Approximating the Traveling Salesperson Problem

Recall in lecture, we learned the following approximation algorithm for TSP:

1. Given the complete graph $G = (V, E)$, compute its MST.
2. Run DFS on the MST computed in the previous step, and record down the the vertices visited in pre-order.
3. The output tour is the list of vertices computed in the previous step, with the first vertex appended to the end.

When G satisfies the triangle inequality, this algorithm achieves an approximation factor of 2. But what happens when triangle inequality does not hold?

Suppose we run this approximation algorithm on the following graph:



The algorithm will return different tours based on the choices it makes during its depth first traversal.

1. Which DFS traversal leads to the best possible output tour?

2. Which DFS traversal leads to the worst possible output tour?

3. What is the approximation ratio given by the algorithm in the worst case for the above instance? Why is it worse than 2?

2 Boba Shops

A rectangular city is divided into a grid of $m \times n$ blocks. You would like to set up boba shops so that for every block in the city, either there is a boba shop within the block or there is one in a neighboring block (assume there are up to 4 neighboring blocks for every block). It costs r_{ij} to rent space for a boba shop in block ij.

Write an integer linear program to determine on which blocks to set up the boba shops, so as to minimize the total rental costs.

(a) What are your variables, and what do they mean?

(b) What is the objective function? Briefly justify.

(c) What are the constraints? Briefly justify.

(d) Solving the non-integer version of the linear program yields a real-valued solution. How would you round the LP solution to obtain an integer solution to the problem? Describe the algorithm in at most two sentences.

(e) What is the approximation ratio of your algorithm in part (d)? Briefly justify.

3 Maximum Coverage

In the maximum coverage problem, we have m subsets of the set $\{1, 2, \ldots n\}$, denoted $S_1, S_2, \ldots S_m$. We are given an integer k, and we want to choose k sets whose union is as large as possible.

Give an efficient algorithm that finds k sets whose union has size at least $(1 - 1/e) \cdot OPT$, where OPT is the maximum number of elements in the union of any k sets. In other words, $OPT = \max_{i_1, i_2, \ldots i_k} |\cup_{j=1}^{k} S_{i_j}|$. Provide an algorithm description and justify the lower bound on the number of elements covered by your solution.

Hint: be greedy! For the proof, you may use the follow property without proof: $(1 - 1/n)^n \leq 1/e$ *for all* $n \in \mathbb{Z}$.