Assignment 7

COMP 4500
Due: April 19, 2024 by 11:59 p.m.
Upload your Word, PDF, or LaTeX file(s) to Blackboard

Exercises from the Textbook (45 points):

Page 651: 1, 3, 5

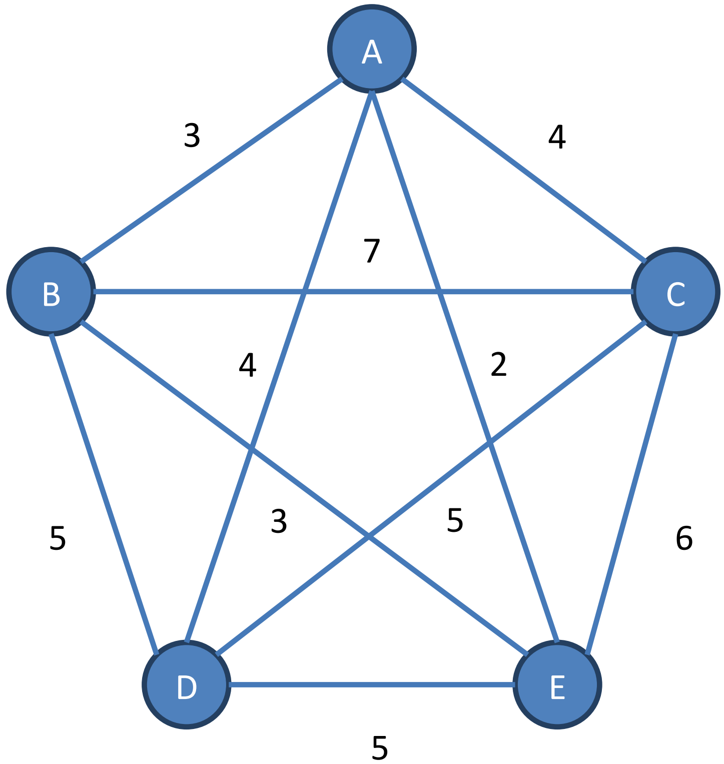
Additional Exercises (55 points):

1. **(15 points)** Consider the Loading Balancing Problem with three machines and seven jobs whose processing times are as follows: 11, 17, 19, 10, 22, 4, 34
    a. State the makespan found using the greedy approximation algorithm given on page 601.
    b. State the makespan found using the improved approximation algorithm given on page 605.
    c. What is the optimal makespan? Is it better than either approximation?

2. **(15 points)** Consider the version of the Traveling Salesman Problem that is an optimization problem on a weighted, undirected graph in which all edge weights are positive integers. In this version, there's an edge between every pair of nodes. The goal is to find a tour (a cycle that visits every node exactly once) with the lowest possible weight. If the graph is metric, meaning that all nodes are connected to all other nodes and those weights obey the triangle inequality, it is possible to find an approximation to the TSP that is no worse than twice the optimal. In other words, if the optimal TSP tour has a length of k, the approximation below will find a tour no longer than $2k$. (There is, in fact, another approximation that finds a tour no longer than $1.5k$, but it's more complicated.)

    To do this approximation, first make a minimum spanning tree of the graph. Next, perform a depth-first search (DFS) of the minimum spanning tree. That DFS will give you a list of nodes. Now, simply visit those nodes in order on the *original* graph, recording the direct edge costs between subsequent nodes.

    Consider the graph on the following page.



a. Make a minimum spanning tree of this graph and show the result.
b. Perform a DFS starting at node A. Record the order of the nodes you visit. Create a tour that follows this order and record the length of the tour.
c. Can you find a tour with smaller weight? If so, what is it?

3. **(10 points)** In order for the approximation algorithm in the previous problem to work, the graph must be metric: All its edges must obey the triangle inequality, and every node must be connected to every other node.
   a. Why do you think this is? **Hint:** The approximation algorithm given is never worse than twice the optimal because it never does worse than twice the length of the MST, and an optimal tour can never be shorter than the MST.
   b. Draw a graph with four nodes whose edge weights do *not* obey the triangle inequality for which the approximation algorithm produces a tour whose weight is more than twice the optimal.

4. **(15 points)** Imagine that a researcher named Polly Gnomielle is able to create an approximation algorithm for the Travelling Salesman Problem that finds a constant approximation in O(n^3) time. Unlike Problem 2 above, this approximation claims to work for *any* weighted, undirected graph in which all nodes are connected to all other nodes, not just those whose edge weights obey the triangle inequality. Although the running time is modest, the algorithm can only guarantee a TSP tour within a factor of 100 of the optimal. In other words, if the optimal TSP tour has a length of k, Dr. Gnomielle's algorithm will find a tour no longer than $100k$.

    a. Another researcher, X. P. Nenshall, is concerned that there might be a problem. If Dr. Gnomielle's approximation algorithm works, Dr. Nenshall thinks it could be possible to solve the NP-Complete problem Hamiltonian Cycle, using the approximation algorithm. For this problem, use the definition of Hamiltonian Cycle as a decision problem that asks whether a tour (a cycle visiting every node exactly once and returning to the starting node) exists in an unweighted, undirected graph. Try to reason how this would be done. How would you add extra edges to a Hamiltonian Cycle instance to turn it into a TSP instance?
    b. The Hamiltonian Cycle problem has unweighted edges, but the TSP uses weighted edges. What weights should you assign to the edges that were already in the Hamiltonian Cycle problem? What weights should you assign to the edges that were **not** in the Hamiltonian Cycle problem? Specifically, what's the lowest weight you can assign to those edges so that the TSP approximation algorithm allows you to find a Hamiltonian Cycle? Note that the weights must be positive integers.
    c. There are two mutually exclusive conclusions you could draw from Dr. Gnomielle's work and Dr. Nenshall's observation about it. What are they?