Homework 4, due Wednesday, November 14

READING: Chapters 17, 19, 22-26.

1. Suppose we perform a sequence of \( n \) operations on a data structure in which the \( i \)-th operation costs \( i \) if \( i \) is an exact power of 2, and 1 otherwise. Use the accounting method to determine the amortized cost per operation and use this to get an upper bound on the actual cost of the sequence of \( n \) operations. (15 points)

2. Consider the same set-up as in the previous problem but now use the potential method to determine the amortized cost per operation and use this to get an upper bound on the actual cost of the sequence of \( n \) operations. (15 points)

3. Exercise 19.4-1 on page 526. (10 points)

4. Consider the following matrix

\[
A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}.
\]

What is \( A^n \) where \( n \) is a positive integer? (10 points)

5. Describe an algorithm for finding a spanning tree with minimum weight containing a specified acyclic (cycle-free) set of edges \( A \) in a connected weighted undirected graph. (10 points)

6. Consider a connected weighted graph where the edge weights are all distinct. Show that the minimum spanning tree must be unique but the second-best minimum spanning tree need not be unique. Is a shortest path between two vertices unique? (15 points)

7. Exercise 24.3-5 on page 663. (10 points)
Let \( G = (V, E) \) be a weighted, directed graph with \( n \) vertices and \( m \) edges, where all edge weights are non-negative. Define the bottleneck distance between two vertices as follows:

\[
\delta_b(x, y) = \min \{ w(p) \mid p \text{ is a path from } x \text{ to } y \},
\]

where \( w(v_0, \ldots, v_k) = \max \{ w(v_{j-1}, v_j) \mid j = 1, \ldots, k \} \).

Give an \( O(m \log n) \) algorithm that computes the bottleneck distance between a source and any other vertex. (15 points)