Solve the following problems from your textbook (Introduction to Algorithms, second edition, by T. Cormen, C. Leiserson, R. Rivest, C. Stein, McGraw Hill).

Problem 1: 17.1-2 Show that if a DECREMENT operation were included in the *k*-bit counter example, *n* operations could cost as much as $\Theta(nk)$ time.

Problem 2: 17.2-1 A sequence of stack operations is performed on a stack whose size never exceeds *k*. After every *k* operations, a copy of the entire stack is made for backup purposes. Show that the cost of *n* stack operations, including copying the stack, is $O(n)$ by assigning suitable amortized costs to the various stack operations.

Problem 3: 22.5-7 A directed graph $G = (V, E)$ is said to be ***semiconnected*** if, for all pairs of vertices $u, v \in V$, we have path from *u* to *v* or from *v* to *u*. Give an efficient algorithm to determine whether or not *G* is semiconnected. Prove that your algorithm is correct, and analyze its running time.

Problem 4: Show that a graph has a unique minimum spanning tree if all the weights of G are distinct.

Problem 5: 24.1-1 Run the Bellman-Ford algorithm on the directed graph of Figure 24.4, using vertex *z* as the source. In each pass, relax edges in the same order as in the figure, and show the *d* and π values after each pass. Now, change the weight of edge (z, x) to 4 and run the algorithm again, using *s* as the source.

Problem 6: 25.2-6 How can the output of the Floyd-Warshall algorithm be used to detect the presence of a negative-weight cycle?

Problem 7: 26.2-2 Show the execution of the Edmonds-Karp algorithm on the flow network of Figure 26.1(a).

Problem 8: 26.3-1 Run the Ford-Fulkerson algorithm on the flow network in Figure 26.8(b) and show the residual network after each flow augmentation. Number the vertices in *L* top to bottom from 1 to 5 and in *R* top to bottom from 6 to 9. For each iteration, pick the augmenting path that is lexicographically smallest