1. Suppose the average case time complexity and amortized time complexity of an operation are O(lgn) and O(n) respectively, where n is the input size. In a sequence of m such operations, which one of the following can be the worst case time complexity of performing all m operations:
   a. O(mlgn)
   b. O(mn)
   c. O(nlgn)
   d. None of the other options

   (Total amortized cost = O(mn) is an upper bound on the actual cost)

2. Suppose that in a k-bit binary counter supporting only increment operations, the counter is not initialized to 0 (so the first increment operation can start at any value). Which one of the following best describes the amortized cost of an increment operation in a sequence of n increment operations in such a counter?
   a. O(n)
   b. O(1)
   c. O(nk)
   d. None of the other options

   (Easy to see. Also discussed in potential method part in text)

3. Consider a stack with a fixed size K with PUSH, POP operations (assume PUSH operations fail in O(1) time if stack is full and POP operations fail in O(1) time if stack is empty). Which of the following are NOT valid potential function for amortized analysis of a sequence of n PUSH and POP operations (choose all that apply)?
   a. The number of elements in the stack
   b. The number of free spaces in the stack
   c. The number of free spaces - the number of elements in the stack
   d. The number of free spaces + the number of elements in the stack

   (Consider a sequence of PUSHs. Options b and c violate the $\phi(D_{i)} \geq \phi(D_0)$ condition for a potential function)

4. Consider a priority queue implemented using a Fibonacci Heap. Suppose we perform n Insert operations (n is a power of 2), and then one ExtractMin operation, in that order. Which of the following is true for the pair of values (no. of root nodes, max. degree of a root node) for the final heap?
   a. (lgn, lgn)
   b. (1, (n-1)/lgn)
   c. (1, lgn)

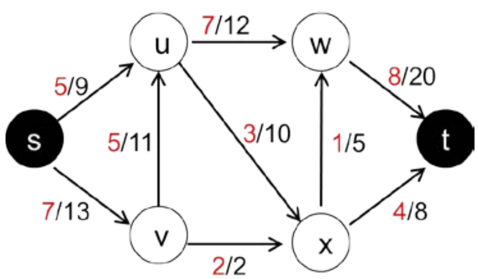
d. (lgn, (n-1)/lgn)

(Inserts will just create a root list with n roots, with no root having any children. ExtractMin will consolidate to one root as n is a power of 2. The degree is easy to see, just work out with n=4, 8 if you are unsure)

5. Which of the following correctly describes the amortized costs of the operations Insert, ExtractMin, Merge, Delete, and DecreaseKey, in that order?
   a. O(1), O(lgn), O(lgn), O(1), O(lgn)
   b. O(1), O(1), O(1), O(lgn), O(lgn)
   c. O(1), O(lgn), O(1), O(1), O(lgn)
   d. O(1), O(lgn), O(1), O(lgn), O(1)

   (Just memorization of the complexities)

6. Consider the following flow network, with x/y on each edge (u,v) indicating f(u,v) = x and c(u,v) = y. Which one of the following sequence defines the augmenting path with the highest residual capacity?



   a. (s, u, v, x, t)
   b. (s, u, w, x, t)
   c. (s, v, u, w, x, t)
   d. (s, v, u, w, t)

   (Just work out, though you do not need to work out all options fully to answer, and it is simple enough to work out just by looking at the picture)

7. Suppose that a maximum flow is already computed in a graph with integral capacities. Then the capacity of exactly one edge is increased by 1. Which of the following is true about the complexity of finding the new maximum flow?
   a. O(1), as the max flow will not change
   b. O(1), as the capacity of only 1 edge is increased by 1
   c. O(V+E)
   d. O(VE)

(The increase in capacity by 1 in 1 edge can increase the maximum flow by at most 1 (if the edge is in the minimum cut). So just start with the existing flow (which is a feasible flow obviously) and do one augmentation if possible in the residual graph. Since capacities are integral, the augmentation will increase the flow by 1 if the flow actually increased)

8. Which of the following statements is true?
   a. If all capacities are even then there is a maximum flow f such that f(u,v) is even for all edges (u,v)
   b. If all capacities are odd, then in any maximum flow f, f(u,v) is odd for all edges (u,v)
   c. If all capacities are integral, then in any maximum flow f, f(u,v) is integral for all edges (u,v)
   d. If flow values in all edges in a maximum flow are integral, then all capacities must be integral

   (Option (a) is true because since capacities are all even, we can divide the capacity by 2 for all edges, which will still give integral capacity for all edges, and so there exists a maximum flow which will assign integral flow to all edges in this new graph. Now just multiply the flows along each edge by 2 to get the maximum flow in the original graph with f(u,v) even for all edges (u,v). That the others are false I leave up to you to show, easy to find simple counterexamples. This was probably the only not-so-easy (given the time) question imho)

9. Which of the following statement is false with regards to the preflow push algorithm?
   a. If a node u has an excess flow, either push or relabel must be applicable to it
   b. If a node u has an excess flow, and a vertex v such that h(u) = h(v) + 1, then push(u,v) can be done
   c. A relabel operation cannot add any new edge to the residual graph
   d. If a node u has an excess and a relabel operation is applied to it, then just after the relabel operation, a push operation is applicable to u.

   (push(u,v) will also need the residual capacity from u to v to be > 0)

10. Suppose that in some intermediate stage of running the preflow-push algorithm, h(u) = 2 and h(v) = 4 for two nodes u, v. What is the value of f(u,v) at that point?
    a. 0 (numerical type, single answer)

    (Suppose it is not 0. Then the edge (v,u) exists in the residual graph. But h(v) = 4 is not ≤ h(u) + 1 as it should be for the height function. This is a contradiction as preflow-push always maintains the height function property. So the edge (v,u) cannot exist)