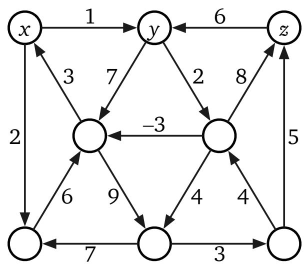
Write your answers in the separate answer booklet.
Please return this question sheet and your cheat sheet with your answers.

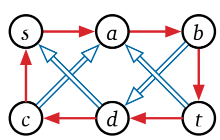
1. ***Clearly*** indicate the following structures in the directed graph below, or write NONE if the indicated structure does not exist. Don't be subtle; to indicate a collection of edges, draw a heavy black line along the entire length of each edge.



   (a) A depth-first tree rooted at x.
   (b) A breadth-first tree rooted at y.
   (c) A shortest-path tree rooted at z.
   (d) The shortest directed cycle.

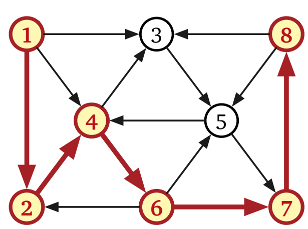
2. Suppose you are given a directed graph G where some edges are red and the remaining edges are blue. Describe an algorithm to find the shortest walk in G from one vertex s to another vertex t in which no three consecutive edges have the same color. That is, if the walk contains two red edges in a row, the next edge must be blue, and if the walk contains two blue edges in a row, the next edge must be red.

   For example, if you are given the graph below (where single arrows are red and double arrows are blue), your algorithm should return the integer 7, because the shortest legal walk from s to t is $s{\rightarrow}a{\rightarrow}b{\Rightarrow}d{\rightarrow}c{\Rightarrow}a{\rightarrow}b{\rightarrow}c$.



3. Let G be an arbitrary (*not* necessarily acyclic) directed graph in which every vertex v has an integer label $\ell(v)$. Describe an algorithm to find the longest directed path in G whose vertex labels define an increasing sequence. Assume all labels are distinct.

   For example, given the following graph as input, your algorithm should return the integer 5, which is the length of the increasing path $1{\rightarrow}2{\rightarrow}4{\rightarrow}6{\rightarrow}7{\rightarrow}8$.



4. Suppose you have an integer array $A[1..n]$ that *used* to be sorted, but Swedish hackers have overwritten k entries of A with random numbers. Because you carefully monitor your system for intrusions, you know *how many* entries of A are corrupted, but not *which* entries or what the values are.

   Describe an algorithm to determine whether your corrupted array A contains an integer x. Your input consists of the array A, the integer k, and the target integer x. For example, if A is the following array, $k = 4$, and $x = 17$, your algorithm should return TRUE. (The corrupted entries of the array are shaded.)

2	3	99	7	11	13	17	19	25	29	31	−5	41	43	47	53	8	61	67	71

   Assume that x is not equal to any of the the corrupted values, and that all n array entries are distinct. Report the running time of your algorithm as a function of n and k. A solution only for the special case $k = 1$ is worth 5 points; a complete solution for arbitrary k is worth 10 points. *[Hint: First consider $k = 0$; then consider $k = 1$.]*

5. Suppose you give one of your interns at Twitbook an undirected graph G with weighted edges, and you ask them to compute a shortest-path tree rooted at a particular vertex. Two weeks later, your intern finally comes back with a spanning tree T of G. Unfortunately, the intern didn't record the shortest-path distances, the direction of the shortest-path edges, or even the source vertex (which you and the intern have both forgotten).

   Describe and analyze an algorithm to determine, given a weighted undirected graph G and a spanning tree T of G, whether T is in fact a *shortest-path* tree in G. Assume all edge weights are non-negative.

   For example, given the inputs shown below, your algorithm should return TRUE for the example on the left, because T is a shortest-path tree rooted at the upper right vertex of G, but your algorithm should return FALSE for the example on the right.