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. Let $G$ be a directed graph, where every vertex $v$ has an associated height $h(v)$, and for every edge $u \rightarrow v$ we have the inequality $h(u) > h(v)$. Assume all heights are distinct. The span of a path from $u$ to $v$ is the height difference $h(u) - h(v)$.

Describe and analyze an algorithm to find the minimum span of a path in $G$ with at least $k$ edges. Your input consists of the graph $G$, the vertex heights $h(\cdot)$, and the integer $k$. Report the running time of your algorithm as a function of $V$, $E$, and $k$.

For example, given the following labeled graph and the integer $k = 3$ as input, your algorithm should return the integer 4, which is the span of the path 8→7→6→4.

3. 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.)

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 $k = 1$.]
4. Suppose you are given a directed graph $G$ in which every edge is either red or blue, and a subset of the vertices are marked as special. A walk in $G$ is legal if color changes happen only at special vertices. That is, for any two consecutive edges $u \rightarrow v \rightarrow w$ in a legal walk, if the edges $u \rightarrow v$ and $v \rightarrow w$ have different colors, the intermediate vertex $v$ must be special.

Describe and analyze an algorithm that either returns the length of the shortest legal walk from vertex $s$ to vertex $t$, or correctly reports that no such walk exists.\(^1\)

For example, if you are given the following graph below as input (where single arrows are red, double arrows are blue), with special vertices $x$ and $y$, your algorithm should return the integer 8, which is the length of the shortest legal walk $s \rightarrow x \rightarrow a \rightarrow b \rightarrow x \Rightarrow y \Rightarrow b \Rightarrow c \Rightarrow t$. The shorter walk $s \rightarrow a \Rightarrow c \Rightarrow t$ is not legal, because vertex $b$ is not special.

5. Let $G$ be a directed graph with weighted edges, in which every vertex is colored either red, green, or blue. Describe and analyze an algorithm to compute the length of the shortest walk in $G$ that starts at a red vertex, then visits any number of vertices of any color, then visits a green vertex, then visits any number of vertices of any color, then visits a blue vertex, then visits any number of vertices of any color, and finally ends at a red vertex. Assume all edge weights are positive.

\(^1\)If you’ve read China Miéville’s excellent novel *The City & the City*, this problem should look familiar. If you haven’t read *The City & the City*, I can’t tell you why this problem should look familiar without spoiling the book.