1. Clearly indicate the following spanning trees in the weighted graph pictured below. Some of these subproblems have more than one correct answer.

(a) A depth-first spanning tree rooted at \(s\)
(b) A breadth-first spanning tree rooted at \(s\)
(c) A shortest-path tree rooted at \(s\)
(d) A minimum spanning tree
(e) A maximum spanning tree

2. A polygonal path is a sequence of line segments joined end-to-end; the endpoints of these line segments are called the vertices of the path. The length of a polygonal path is the sum of the lengths of its segments. A polygonal path with vertices \((x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)\) is monotonically increasing if \(x_i < x_{i+1}\) and \(y_i < y_{i+1}\) for every index \(i\)—informally, each vertex of the path is above and to the right of its predecessor.

Suppose you are given a set \(S\) of \(n\) points in the plane, represented as two arrays \(X[1..n]\) and \(Y[1..n]\). Describe and analyze an algorithm to compute the length of the maximum-length monotonically increasing path with vertices in \(S\). Assume you have a subroutine \(\text{LENGTH}(x, y, x', y')\) that returns the length of the segment from \((x, y)\) to \((x', y')\).
3. Suppose you are maintaining a circular array $X[0..n-1]$ of counters, each taking a value from the set $\{0,1,2\}$. The following algorithm increments one of the counters; if the counter overflows, the algorithm resets it 0 and recursively increments its two neighbors.

\begin{verbatim}
INCREMENT(i):
  X[i] ← X[i] + 1
  if X[i] = 3
    X[i] ← 0
    INCREMENT((i - 1) mod n)
    INCREMENT((i + 1) mod n)
\end{verbatim}

(a) Suppose $n = 5$ and $X = [2, 2, 2, 2, 2]$. What does $X$ contain after we call $INCREMENT(3)$?
(b) Suppose all counters are initially 0. Prove that $INCREMENT$ runs in $O(1)$ amortized time.

4. A looped tree is a weighted, directed graph built from a binary tree by adding an edge from every leaf back to the root. Every edge has non-negative weight.

![A looped tree.](image)

(a) How much time would Dijkstra's algorithm require to compute the shortest path from an arbitrary vertex $s$ to another arbitrary vertex $t$, in a looped tree with $n$ vertices?
(b) Describe and analyze a faster algorithm. Your algorithm should compute the actual shortest path, not just its length.

5. Consider the following algorithm for finding the smallest element in an unsorted array:

\begin{verbatim}
RANDOMMIN(A[1..n]):
  min ← ∞
  for i ← 1 to n in random order
    if A[i] < min
      min ← A[i]  (*)
  return min
\end{verbatim}

Assume the elements of $A$ are all distinct.

(a) In the worst case, how many times does $RANDOMMIN$ execute line $(\ast)$?
(b) What is the probability that line $(\ast)$ is executed during the last iteration of the for loop?
(c) What is the exact expected number of executions of line $(\ast)$?