1. Describe and analyze an algorithm to find the second smallest spanning tree of a given graph $\mathcal{G}$; that is, the spanning tree of $\mathcal{G}$ with smallest total weight except for the minimum spanning tree.

2. Let S be a set of n points in the plane. A point p in S is called *maximal* if no other point in S is both above and to the right of p. If each point in s is chosen independently and uniformly at random from the unit square $[0,1]\times[0,1]]$, what is the exact expected number of maximal points in S?

3. A *data stream* is an extremely long sequence of items that you can only read in one pass (i.e., each item can only be read once).

   (a) Describe and analyze an algorithm that chooses one element uniformly at random from a data stream, without knowing the length of the stream in advance. Your algorithm should spend $O(1)$ time per stream item and use $O(1)$ space, not including the stream itself (i.e., the space requirement should not depend on the length of the stream).

   (b) Describe and analyze an algorithm that chooses k elements uniformly at random from a data stream, without knowing the length of the stream in advance. Again, your algorithm should spend $O(1)$ time per stream item and use $O(k)$ space, not including the stream itself.

4. Consider the Minimum Spanning Tree Problem on an undirected graph. $G = (V, E)$, with a cost $c_e \geq 0$ on each edge, where the costs may not all be different. If the costs are not all distinct, there can in general be many distinct minimum-cost solutions. Suppose we are given a spanning tree T with the guarantee that for every $e \in T$, e belongs to some minimum-cost spanning tree in G. Can we conclude that T itself must be a minimum-cost spanning tree in G? Give a proof or a counterexample with explanation.