1. Select all the problems for which we know of an efficient polynomial time algorithm through techniques studied thus far in this specialization. **4 / 4 points**

- [x] Given a weighted graph G, does there exist a negative weight cycle?

  Correct
  We studied the Bellman Ford algorithm that can solve this problem.

- [x] Given a graph is there a spanning tree whose weight is less than K?

  Correct
  We can use the MST algorithm to compute the minimum weight spanning tree and using the weight of this spanning tree, we can easily answer the question in polynomial time

- [x] Given an array `a` of size `n` and a number `k`, are there more than `n/4` elements which are `>=` k?

  Correct
  This is polynomial time solvable: simply use k as a pivot and partition the array. You can answer the question based on the sizes of the two partitions thus created.

- [ ] Given a graph G and two vertices s, t, is the longest simple path (a path that does not visit any vertex more than once) of length more than k?

2. Select all the correct answers regarding certificates from the list below. **1 / 1 point**

- [x] Given a number n that is known to be a product of two large prime numbers (such numbers are used extensively in cryptography), we wish to find out the kth bit of its smallest prime factor. The certificate is the prime factor p itself.

  Correct
  Correct: we can check that the certificate is a prime factor: we can check that it is a factor of n, we can check that $n/p \geq p$ and finally, we can extract the kth bit of p, to check the answer.

- [x] Let G be a weighted directed graph and s,t be two vertices of G. We wish to know if there a path from s to t of length $\leq W$. The empty string can be a certificate for this problem that can be checked in polynomial time.

  Correct
  Correct: simply run Dijkstra's algorithm and it will give us the shortest path weight from s to t. We do not need a certificate since Dijkstra's algorithm can run in polynomial time.

- [x] Let G be an undirected graph. We wish to know if there is a cycle that visits all vertices of G exactly once. The certificate for a yes answer is given by the cycle itself

  Correct
  Correct: the cycle can be verified in polynomial time in the size of the graph: check that it is a cycle, check that it visits each vertex exactly once.

- [ ] We are given an instance of the knapsack problem weights $W_1, \ldots, W_n$ and values $V_1, \ldots, V_n$. We wish to know if we can select items with total weight $\leq W$ and value $\geq v$. Suppose the algorithm comes back with the answer "no". The certificate for this answer is just a selection of items whose weight is $\leq W$ and value $< v$