Problem 1: For each statement below, say if it is True or False.

- Dijkstra’s algorithm works on graphs with arbitrary weight functions. False
- The Prim algorithm to compute the MST of a graph is a dynamic programming algorithm False
- The Floyd-Warshall algorithm to compute all pairs shortest paths in a graph is a greedy algorithm False
- Multiplying two polynomials of degree \( n \) can be done \( O(n \log n) \) scalar multiplications True
- If you run Euclid’s algorithm to compute the GCD of two integers \( a, b \), the algorithm will make \( \Theta(\log b) \) recursive calls. True

Problem 2: Amortized Analysis. A sequence of \( n \) operations is performed on a data structure. The \( i^{th} \) operation costs \( i \) if \( i \) is a perfect square, otherwise it costs 1. Determine the total cost of the sequence of operations, and the amortized cost per operation.

Problem 3: The square of a directed graph \( G = (V, E) \) is the graph \( G^2 = (V, E^2) \) such that \((u, w) \in E^2\) if and only if there exists a \( v \in V \) such that \((u, v) \in E\) and \((v, w) \in E\). In other words \( G^2 \) contains an edge between \( u \) and \( w \) whenever \( G \) contains a path from \( u \) to \( w \) of length exactly 2.

Show how to compute \( G^2 \) when given \( G \) in either adjacency-list or adjacency-matrix representation.

Problem 4: Prove or disprove: if a directed graph \( G \) contains cycles then the DFS-based topological sort algorithm described in class (and in Section 22.4 of the book) produces a vertex ordering that minimizes the number of “bad” edges, i.e. edges that are inconsistent with the ordering produced by the algorithm.

Problem 5: When using the complex root of unity to compute the FFT, one might encounter problems related to precision errors. When dealing with polynomials with only integer coefficients, it might be desirable to use different interpolation points that are not subject to precision errors. One possible approach uses modular arithmetic. Let \( p \) be a prime such that \( p = kn + 1 \) where \( n \) is the length of the FFT input vector and \( k \) is arbitrary constant (chosen as small as possible). Assume that such prime can be found easily.

As we saw in class then for any element \( a \in \mathbb{Z}_p^* \) we have that \( a^{p-1} = a^{kn} = 1 \mod p \). Consider an element \( g \in \mathbb{Z}_p^* \) such that \( g^j \neq 1 \mod p \) for all \( 1 \leq j < kn \) (i.e. \( g^j = 1 \mod p \) only if \( j = kn \) – such elements are called generators of \( \mathbb{Z}_p^* \) and can be found efficiently).

Let \( \omega = g^k \mod p \). Note that \( \omega \neq 1 \mod p \) but \( \omega^n = 1 \mod p \), so \( \omega \) is an \( n \)-root of unity in \( \mathbb{Z}_p^* \).

- Let \( p = 17 \). Check that 3 is a generator for \( \mathbb{Z}_{17}^* \), and then compute the \( 8 \)-root of unity in \( \mathbb{Z}_{17}^* \).
- For general \( p \), argue that you can still compute FFT in \( \mathbb{Z}_p^* \) using the divide-and-conquer algorithm we saw in class.

Problem 6: Let \( n \) be an integer, and \( a, b \in \mathbb{Z}_n^* \). If \( GCD(a, n) = 1 \) how many solution does the equation \( ax = b \) have in \( \mathbb{Z}_n^* \)? Justify your answer.