Problem 1 (10 points) Solve exercise 4 in Chapter 2 (pages 67–68) of “Algorithm Design” by Kleinberg and Tardos. Provide a proof for your solution. There are 21 pairs of functions. You need not compare every pair of functions in your proof. By judicial choice of which functions to compare, the number of function pairs you need to compare is much smaller than 21.

1. Keep in mind that in order to prove that \( f(n) \) is \( O(g(n)) \), you must exhibit the constants required by the definition of asymptotic upper bound, and show that the inequality in the definition is satisfied by the functions, given these constants.

2. Note that \( g_4(n) \) is printed before \( g_3(n) \) in the textbook.

Problem 2 (10 points) Solve exercise 5 in Chapter 2 (page 68) of “Algorithm Design” by Kleinberg and Tardos. If you decide that a statement is true, provide a short proof.

Problem 3 (10 points) Let \( A \) and \( B \) be two sets of \( n \) points on a 2-dimensional Euclidean plane. For any point \( a \in A \) and points \( b, b' \in B \), \( a \) prefers \( b \) over \( b' \) if and only if \( \|ab\| < \|ab'\| \). Similarly, for any two points \( a, a' \in A \), \( b \) prefers \( a \) over \( a' \) if and only if \( \|ab\| < \|ab'\| \). Here \( \|ab\| \) is the Euclidean distance between points \( a \) and \( b \). For simplicity, we assume that no two points are at the same distance from any given point.

Consider the following simple matching algorithm. Let \( A_F, B_F \) be the set of free (or unmatched) points of \( A \) and \( B \) respectively. Initially, we set \( M^* \) to \( \emptyset \), \( A_F \leftarrow A \) and \( B_F \leftarrow B \). We repeat the following for \( n \) iterations. In each iteration, we compute \( (a, b) = \arg \min_{a \in A_F, b \in B_F} \|ab\| \). We match \( a \) to \( b \), i.e., add \( (a, b) \) to \( M^* \). Then, we set \( A_F \leftarrow A_F \setminus \{a\} \) and \( B_F \leftarrow B_F \setminus \{b\} \).

At the end of \( n \) iterations, we return \( M^* \). Show that \( M^* \) is a stable perfect matching.