William Romine

Analysis of Algorithms

Dr. Lewis

April 19, 2024

Assignment 08

2

The best performance for a convex hull algorithm is O (n log(n)), which is due to the problem reduction from sorting a set of numbers to computing the convex hull of points in a plane. By mapping each number x to a point (x, x^2), we create a convex parabola where every point must be on the convex hull, and neighboring points on the hull have neighboring x values. This reduction implies that sorting the input numbers is an essential part of the problem, and since sorting has a lower bound of Ω (n log(n)), the convex hull problem also has a lower bound of Ω (n log(n)).

3

1. Convert the planar graph to a dual graph: For a planar graph, create a dual graph by placing a vertex in each face of the original graph and connecting two vertices if their corresponding faces share an edge in the original graph.
2. Transform the dual graph to a weighted graph: Assign weights to the edges of the dual graph based on the number of edges in the shared boundary between the corresponding faces in the original graph.
3. Find a minimum weight perfect matching in the dual graph: A perfect matching is a set of edges such that every vertex is incident to exactly one edge in the set. The minimum weight perfect matching can be found using an algorithm like the Hungarian algorithm or Blossom algorithm.
4. Construct the shortest tour in the original graph: For each edge in the minimum weight perfect matching, add the corresponding edge in the original graph to the tour. Since the dual graph is also planar, the tour will visit each edge of the original graph at least once.
5. Calculate the maximum cut: The number of edges in the maximum cut is equal to the sum of the weights of the edges in the shortest tour plus the number of vertices in the original graph.

4

The brute force algorithm for checking if a number n is composite iterates from 2 to n/2, checking for divisibility. This has a running time of O (n * log n), which is not polynomial in the size of n, so the composite number problem is not in P. It is in co-P, the

complement of P. This problem is about determining if a number has factors within a given range, and the brute force algorithm is to check each number in that range to see if it is a factor. This has a high time complexity and is not an efficient solution for large values of n.