COL215 SW Assignment 1 - Gate Packing

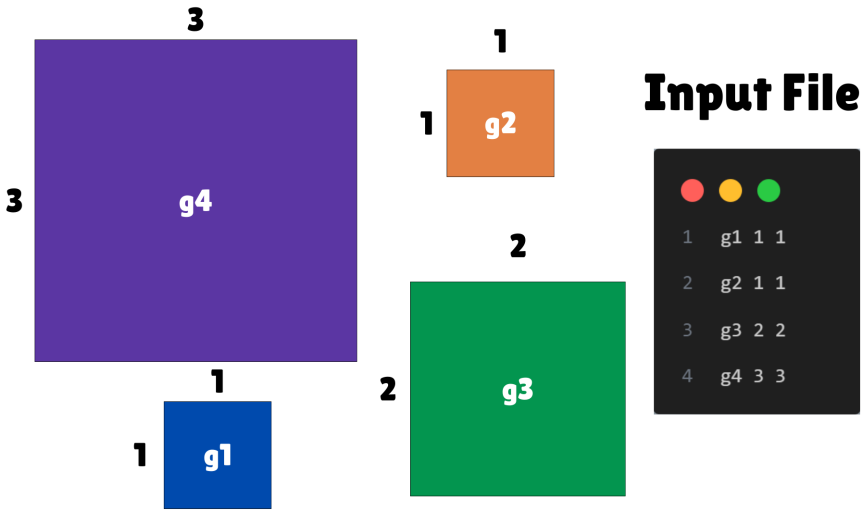
Submission By :

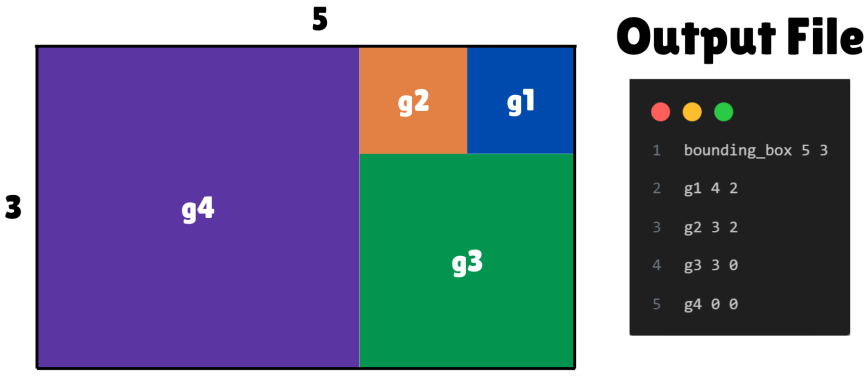
Yash Rawat **2023CS50334**

Priyanshi Gupta **2023CS10106**

Department of Computer Science and Engineering
Indian Institute of Technology, Delhi

August 12, 2024





1 Modelling Gate Packing

1.1 What is Gate Packing ?

In the context of gate level circuit designing refers to the process of arranging logic gates on a circuit board in order to minimize wasted space, reduce interconnection length and optimize the overall layout.

Generalised gate packing is a very complex problem and involves multiple challenges such as placement and routing complexity, heat dissipation, design constraints due to fabrication processes, etc. but we will be tackling a simplified problem in this assignment.

1.2 Understanding the Problem Statement

The problem statement models the gates as a set of n rectangles (provided as input for each test case) : $\{g_1, g_2, ..., g_n\}$ each represented a pair of integers : $g_i = (w_i, h_i)$, where w_i and h_i are the width and the height of the i^{th} board. A given set of gates is said to be "correctly assigned" if no two gates have a overlapping area. The bounding rectangle is defined to be the smallest rectangle that encloses all gates and has the minimum area (out of all the possible "correctly assigned" cases).

The program is supposed to output 2 things - The w and h of the bounding rectangle and the set of coordinates : $\{(x_i, y_i)\}_{i=1}^{i=n}$ - where (x_i, y_i) denote the coordinate of the bottom left corner of the g_i. A sample test case is given below : (Note that every gate placed is in the original orientation provided by test case, i.e. re-orientation by rotation is disallowed)

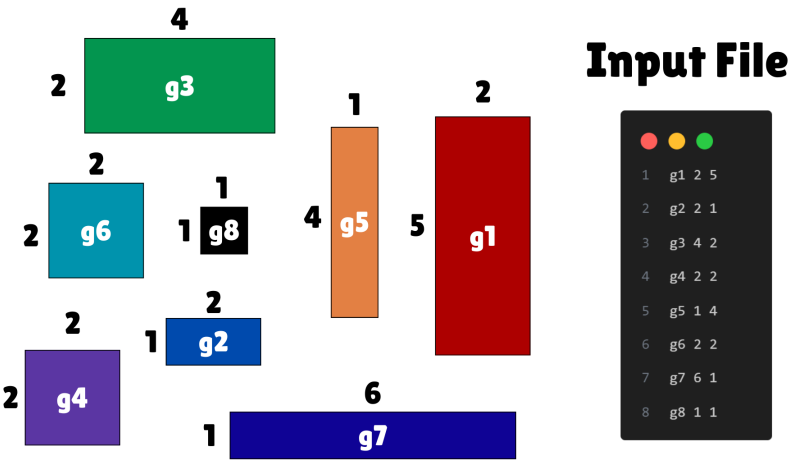


Figure 1: Sample Test Case with 8 gates

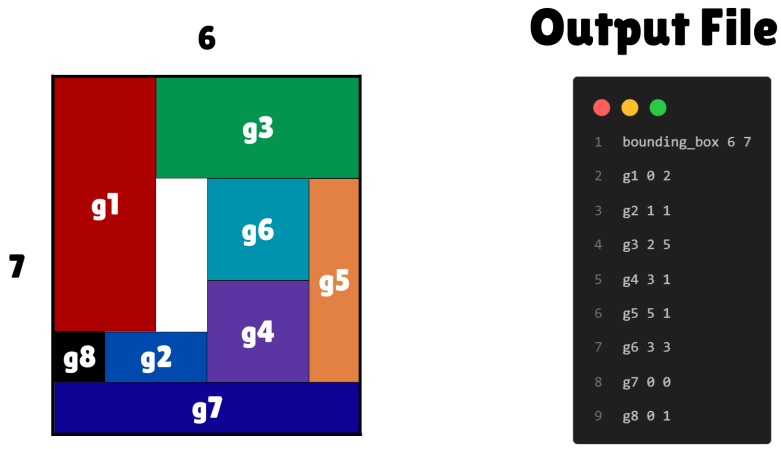


Figure 2: Output of above sample case

2 Design Philosophy

2.1 Design Process Flowchart

2.2 Algorithm Conceptualization & Design

2.3 Proving Correctness of algorithm

3 Time Complexity Analysis

3.1 Elementary Operations

4 Visualising Output on Multiple Test Cases

4.1 Test Case 1

4.2 Test Case 2

4.3 Test Case 3