Pranjit Kumar Kalita
Project #3

Packing of Rectangles using Binary Trees					
File Name	Elapsed Time for Packing Operations	Width of Packing Rectangle	Height of Packing Rectangle	X-coordinate of Largest Indexed Rectangle	Y-coordinate of Largest Indexed Rectangle
r0_flr.txt	0.000000e+00	6.488700e+04	9.299900e+04	0.000000e+00	0.000000e+00
r1_flr.txt	0.000000e+00	1.663549e+06	1.812356e+06	3.707700e+04	0.000000e+00
r2_flr.txt	0.000000e+00	3.411589e+06	4.923995e+06	4.238000e+03	0.000000e+00
r3_flr.txt	0.000000e+00	9.793506e+06	2.573651e+06	9.637800e+05	1.857540e+05
r4_flr.txt	0.000000e+00	9.653391e+06	1.018544e+07	7.987561e+06	3.441761e+06
r5_flr.txt	0.000000e+00	2.745456e+07	1.320092e+07	2.447639e+07	1.683258e+06
r6_flr.txt	0.000000e+00	1.100000e+01	1.500000e+01	9.000000e+00	0.000000e+00

<u>Space Complexity of my packing algorithm</u>: **O(n)**, since whenever I encounter an internal node, I recursively call the function to calculate coordinates. We know that there are (n/2) internal nodes, thus, the space complexity would be O(n). For the function calculating the width and height of each internal node, the space complexity is O(1) since I am not allocating any new memory by making recursive function calls. Therefore, the overall space complexity of my packing algorithm is O(n).

<u>Time Complexity of my packing algorithm</u>: **O(n)**, since I am calculating the coordinates only for the leaf nodes, which are (n/2) in number, thereby leading to an O(n) time complexity. Also, for the function calculating the width and height of each internal node, the time complexity is O(n), since I am only performing calculations for the internal nodes, which are (n/2) in number. Therefore, the overall time complexity is O(n).