Kevin Lee
CS 331

Project Report - Optimal Binary Search Tree

The goal of this project was to implement a memory function for the Optimal Binary Search Tree problem in either Java or Python where the inputs are a probability vector and a keys vector. I chose to create the Optimal Binary Search Tree in Java. The algorithm will find an optimal tree for each set of probabilities and keys. The output is the average key comparisons for the optimal tree and of course, the tree itself in preorder.

I created the program so that the user will be asked how many keys are going to be entered, what the keys are, and what the corresponding probabilities are. For example, the input of the keys would be: don isabelle ralph wally, as designated in the project outline. The corresponding probabilities would be in decimal double notation: .375 .375 .125 .125. After receiving this information from the user, I would then put them into their respective arrays and start the calculations. I have split the algorithm into 4 different methods to handle parts of the job. The algorithm calculates the the minimum value with respect to k and assigns the value of k to its rightful location in the r array. The findMin() is responsible for dealing with find the minimum. The r array is used later to print out the tree in preorder by using printTree(). My results and output are given in the ‘projectOutput’ text file I’ve provided.