**CSE225 Data Structures, 2020(FALL)**

**Project #1 REPORT**

**Author: Engin Bektaş**

**ID: 150118501**

**Text Representation with Binary Trees**

*a) Build a BST with the key “Word”.*



*b) Suppose that the number of accesses to word in your tree is directly the frequency of the word given in the table. Calculate Total Access Time in the tree you build in (a).*

*Σ\_(n = 1 -> size) depth \* frequency =*





*c) Suppose that the number of accesses to word in your tree is directly the frequency of the word given in the table. Construct a BT to keep these records in the main memory so as to minimize the total access time, where one time unit is the time taken to compare the key of a tree with the key searched!*



*d) Calculate Total Access Time in the tree you build in (c).*



*e) Discuss your results in (b) and (d).*

*In (b), our binary tree is designed to be a binary search tree by alphabetic order. Nodes' frequencies are completely arbitrary among depth levels. This makes the calculation of the total access numbers unpredictable.*

*In (d), our binary tree is sorted by nodes' frequencies in level order fashion. From top to bottom, and from left to right, frequencies drop. Nodes with biggest frequencies are lined at top. This configuration of tree is the most efficient for having the minimum total access time.*