Grant Mercer

Assignment 6

Assignment 6 Part B:

Summary of ***trie*** *data structure:*

The trie is an extremely useful data structure when it comes to word related algorithms and problems. This data structure is similar to a binary tree, but as opposed to only having 2 children allowed per parent a trie node may have 26. Each Node represents a letter of the alphabet and can be used to create links between words and avoid unneeded repetition for large amounts of words.

Compare using a ***trie*** to other data structure:

The trie has some key advantages and disadvantages over other traditional data structure. Using a trie with large amount of data can be much more efficient than placing data into a sorted array or avl tree, as trie nodes can be a part of multiple pieces of data. When posed with large amounts of related data, a trie has a key advantage in saving memory. But the data structure also can suffer with lookup times. Unlike an avl tree, there is no guarantee of balance in a trie making a lookup no longer O(log n). It also does not have random access as sorted arrays do.

Big-O for ***trie***operations:

Insertion: O(m) where m is the length of the key

Search(same as sort): O(m) where m is the length of the key

isPrefix(same as sort): O(m) where m is the length of the key