

1)

- Construction of the input tree would be $O(4^n)$ because each node has 4 possible children (i.e, input of "7 9" would give you level one with 4 nodes, level 2 would give each previous node 4 additional nodes.
- Time complexity of java's built in HashSet function is $O(1)$. Time complexity of our Trie method is $O(26 \log n)$ because the key is of size 26 (A-Z) and a general tree's time complexity is $O(\log n)$.
- The time complexity of solving the whole problem is the largest time complexity of all its components, which means the time complexity of the program is $O(4^n)$

2)

- Exhaustive search is when a program must examine every possible alternative to find a solution. Finding all possible words from the keypad input is Exhaustive search.

3)

- Branch and Bound is a method of pruning the tree during its construction. This method will remove all non-word paths when they are created and leave only possible words within the data structure. This greatly improves the speed of a program that will reference the data structure many times, but does little to improve performance on a one time use data structure or the creation of the structure itself.

4)

- You can apply branch and bound to our tree made based on keypad input utilizing the prefix tree as a check for if the current word being formed for keypad is the prefix of a real word in our dictionary tree. You can do this by taking the current path being built in the keypad and finding a path in the prefix tree that has the same letters in the same order. If the prefix tree contains these letters as the start of a word, continue forming that path in the keypad tree. If the prefix tree does not contain the letters as the start of a word, stop adding letters to the path and remove (prune) it from the tree. This will greatly improve the speed of tree traversal and word finding for the program, most notably with large words (i.e., a deep keypad tree)

5)

Five screen shots with different inputs

```
0 = exit program
1 = finish input
2 = 'a''b''c'
3 = 'd''e''f'
4 = 'g''h''i'
5 = 'j''k''l'
6 = 'm''n''o'
7 = 'p''q''r''s'
8 = 't''u''v'
9 = 'w''x''y''z

Remember, when finished with input press 1
if you wish to exit program, press 0

Input one number at a time : 0 0 0 0
Found words for user input include using Hashing:
add
ade
bed
bee
Found words for user input include using Prefix Tree:
add
ade
bed
bee

0 = exit program
1 = finish input
2 = 'a''b''c'
3 = 'd''e''f'
4 = 'g''h''i'
5 = 'j''k''l'
6 = 'm''n''o'
7 = 'p''q''r''s'
8 = 't''u''v'
9 = 'w''x''y''z

Remember, when finished with input press 1
if you wish to exit program, press 0

Input one number at a time : 0
Exiting...

Process finished with exit code 0

0 = exit program
1 = finish input
2 = 'a''b''c'
3 = 'd''e''f'
4 = 'g''h''i'
5 = 'j''k''l'
6 = 'm''n''o'
7 = 'p''q''r''s'
8 = 't''u''v'
9 = 'w''x''y''z

Remember, when finished with input press 1
if you wish to exit program, press 0

Input one number at a time : 0 0 0 0 0 0
Found words for user input include using Hashing:
there
these
Found words for user input include using Prefix Tree:
there
these

0 = exit program
1 = finish input
2 = 'a''b''c'
3 = 'd''e''f'
4 = 'g''h''i'
5 = 'j''k''l'
6 = 'm''n''o'
7 = 'p''q''r''s'
8 = 't''u''v'
9 = 'w''x''y''z

Remember, when finished with input press 1
if you wish to exit program, press 0

Input one number at a time : 0
Exiting...

Process finished with exit code 0

0 = exit program
1 = finish input
2 = 'a''b''c'
3 = 'd''e''f'
4 = 'g''h''i'
5 = 'j''k''l'
6 = 'm''n''o'
7 = 'p''q''r''s'
8 = 't''u''v'
9 = 'w''x''y''z

Remember, when finished with input press 1
if you wish to exit program, press 0

Input one number at a time : 0 0 0 0 0 0
Found words for user input include using Hashing:
hello
Found words for user input include using Prefix Tree:
hello

0 = exit program
1 = finish input
2 = 'a''b''c'
3 = 'd''e''f'
4 = 'g''h''i'
5 = 'j''k''l'
6 = 'm''n''o'
7 = 'p''q''r''s'
8 = 't''u''v'
9 = 'w''x''y''z

Remember, when finished with input press 1
if you wish to exit program, press 0

Input one number at a time : 0
Exiting...

Process finished with exit code 0

0 = exit program
1 = finish input
2 = 'a''b''c'
3 = 'd''e''f'
4 = 'g''h''i'
5 = 'j''k''l'
6 = 'm''n''o'
7 = 'p''q''r''s'
8 = 't''u''v'
9 = 'w''x''y''z

Remember, when finished with input press 1
if you wish to exit program, press 0

Input one number at a time : 0 0 0 0 0 0
Found words for user input include using Hashing:
wilson
Found words for user input include using Prefix Tree:
wilson

0 = exit program
1 = finish input
2 = 'a''b''c'
3 = 'd''e''f'
4 = 'g''h''i'
5 = 'j''k''l'
6 = 'm''n''o'
7 = 'p''q''r''s'
8 = 't''u''v'
9 = 'w''x''y''z

Remember, when finished with input press 1
if you wish to exit program, press 0

Input one number at a time : 0
Exiting...

Process finished with exit code 0

0 = exit program
1 = finish input
2 = 'a''b''c'
3 = 'd''e''f'
4 = 'g''h''i'
5 = 'j''k''l'
6 = 'm''n''o'
7 = 'p''q''r''s'
8 = 't''u''v'
9 = 'w''x''y''z

Remember, when finished with input press 1
if you wish to exit program, press 0

Input one number at a time : 0 0 0 0 0 0
Found words for user input include using Hashing:
apple
Found words for user input include using Prefix Tree:
apple

0 = exit program
1 = finish input
2 = 'a''b''c'
3 = 'd''e''f'
4 = 'g''h''i'
5 = 'j''k''l'
6 = 'm''n''o'
7 = 'p''q''r''s'
8 = 't''u''v'
9 = 'w''x''y''z

Remember, when finished with input press 1
if you wish to exit program, press 0

Input one number at a time : 0
Exiting...

Process finished with exit code 0
```