Trie (Letter Tree)

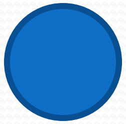
Mostafa Saad

Trie

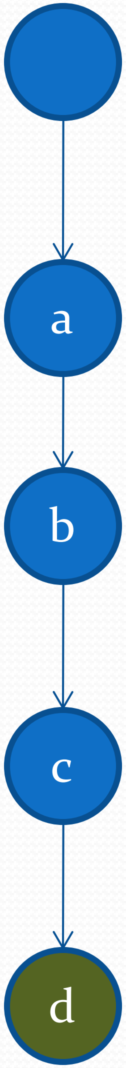
- A simple but powerful data structure.
- It adds/searches for string in O(L)

- It is a tree with branches as letters

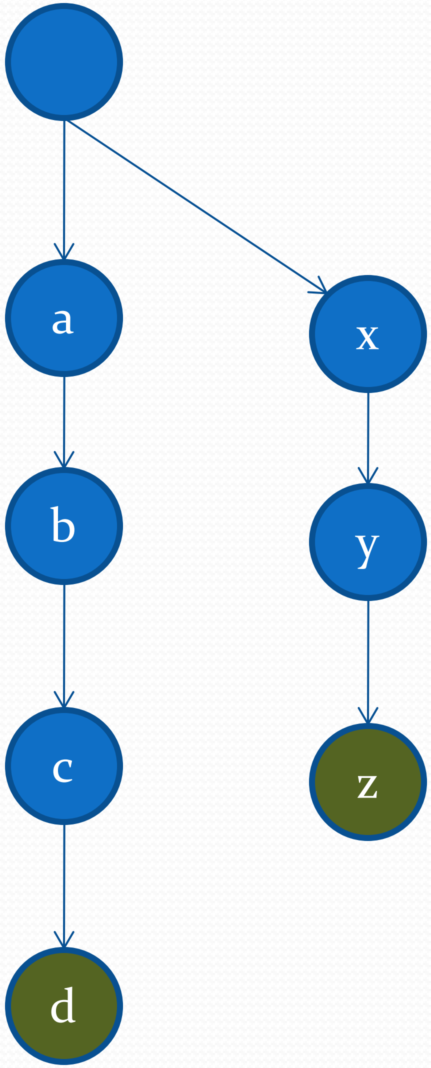
Initially, root exist



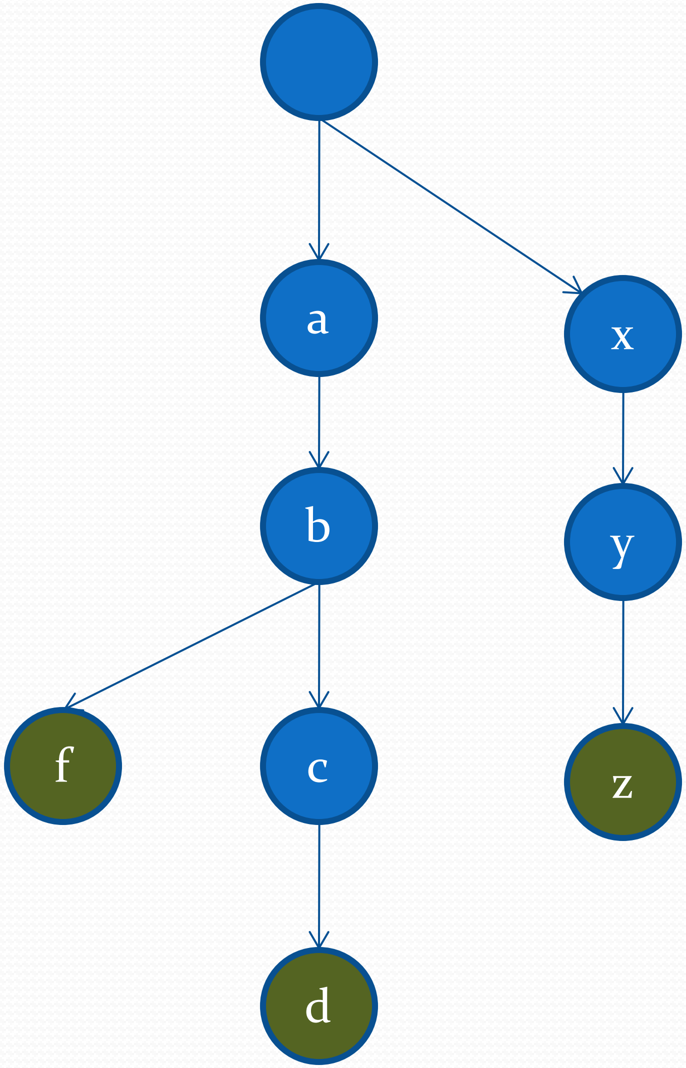
Add abcd



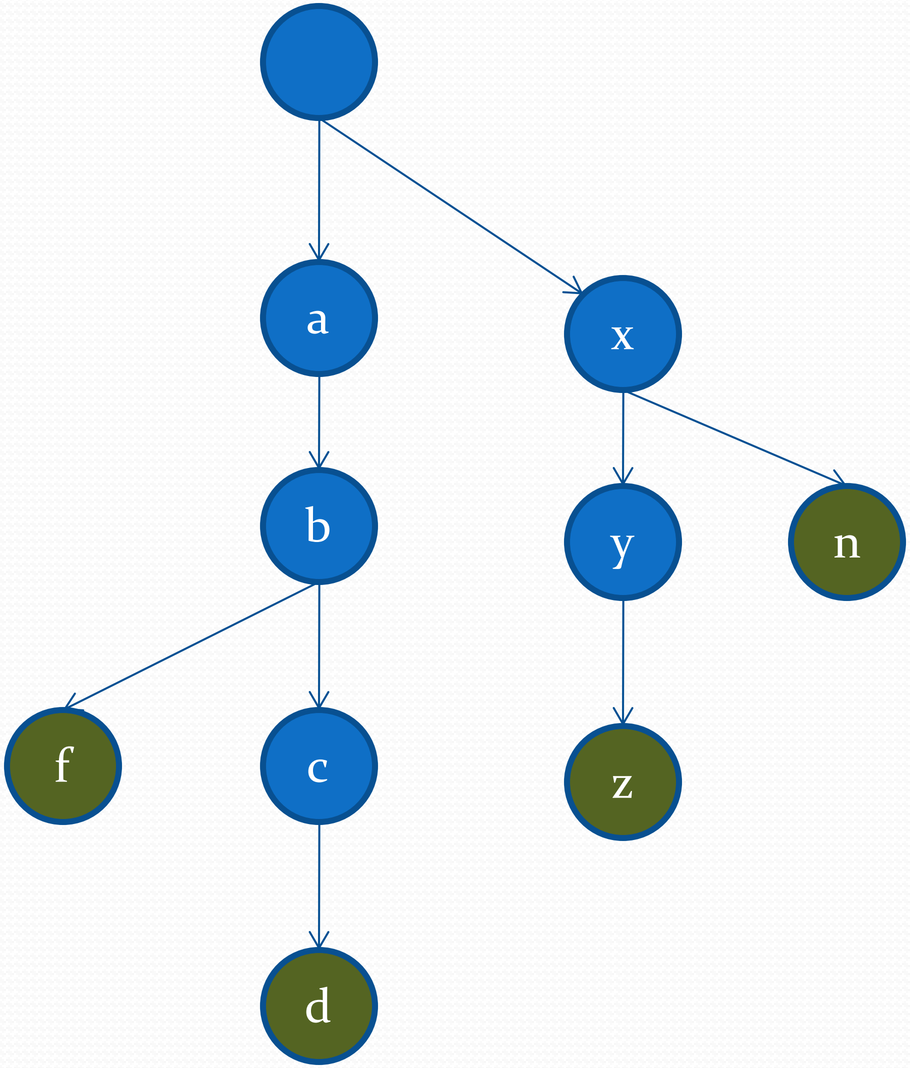
Add xyz



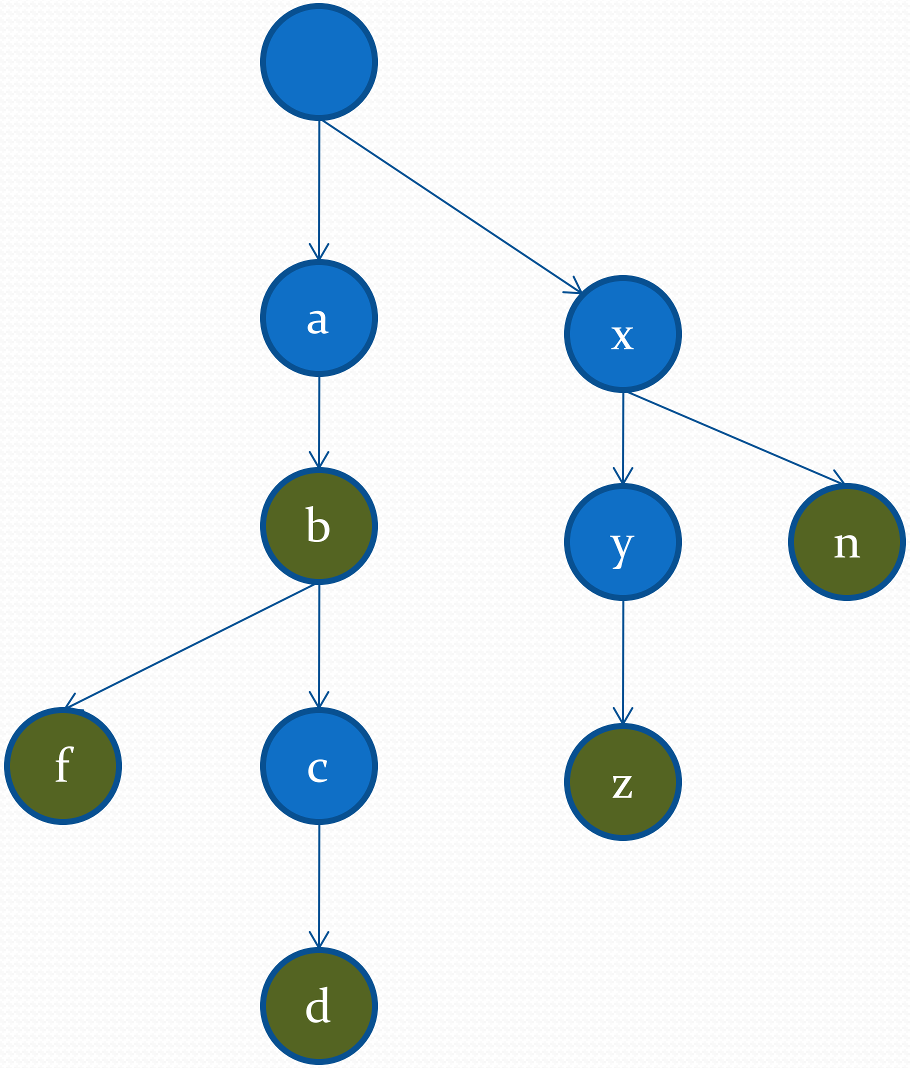
Add abf



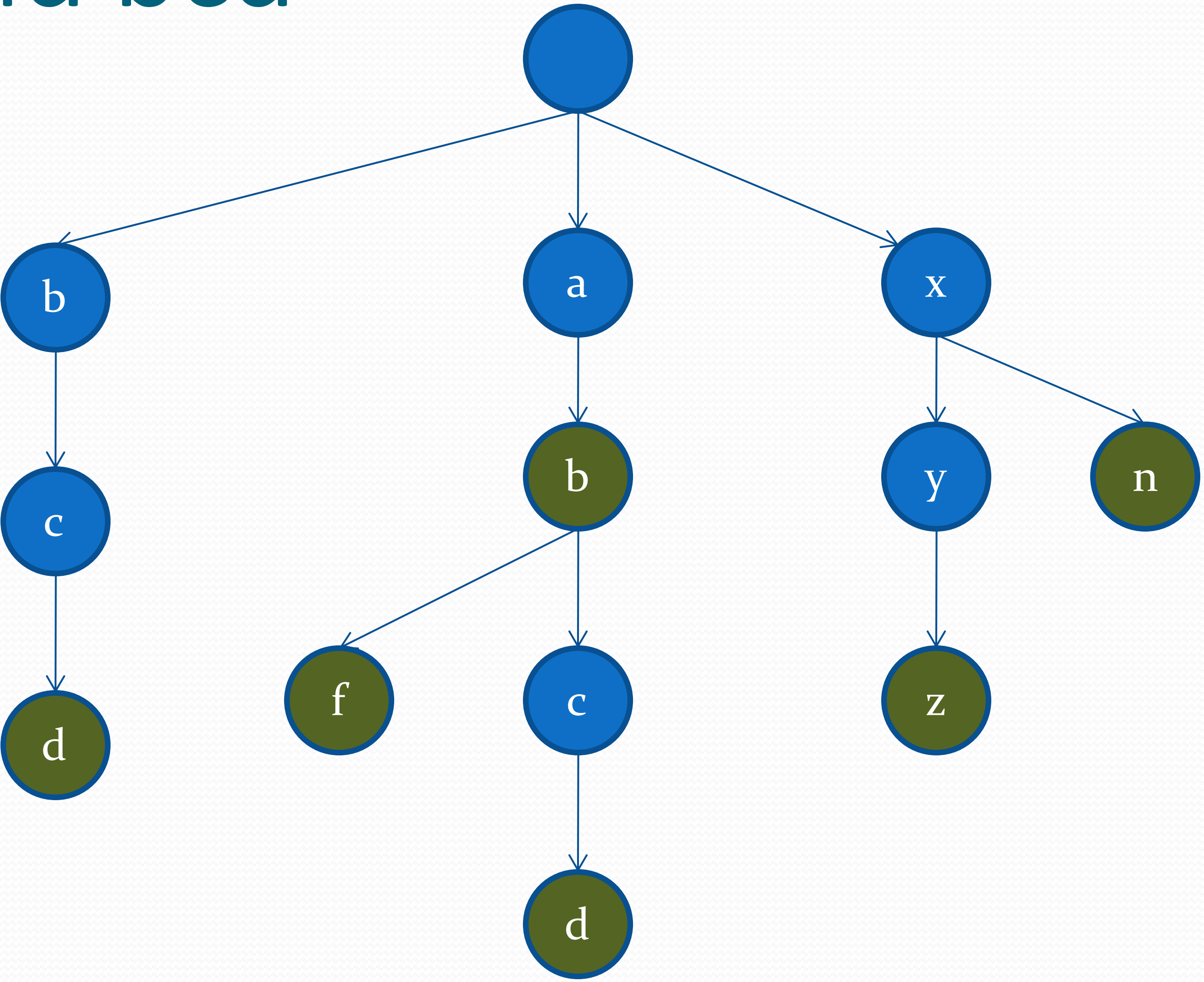
Add xn



Add ab



Add bcd



So

- Simple tree
- Common prefixes are created once
- Easy to know the nodes (red ones)
- Could answer Questions such as:
  - Does word exist? Frequency?
  - Does prefix exist? Frequency?
  - Any question is answered by a trace from root to maximum a leaf
  - So O(L) where L is word length
  - Could also print all strings on O(S), where S # of words

Implementation

- Node is similar to binary search tree, but we have many children
  - Caring with memory?
    - Either have linked list for the set of nodes
    - Or more efficient. Use maps
  - Memory not issue
    - Has an array of nodes, its size the # of characters
- Node could contain whatever needed
  - E.g. boolean to know full words
  - E.g. integer for frequency