Relatório T2 INE5408

Aluno: Jan Bortolanza - 20203632

Struct TrieNode :

Essa estrutura representa um nó da Trie. Cada nó possui um array de ponteiros para os nós filhos, indicando as possíveis letras que podem seguir a partir desse nó. A variável isWord indica se o nó representa o fim de uma palavra. As variáveis position e length armazenam a posição inicial e o comprimento da palavra no arquivo do dicionário. A variável wordCount conta o número de palavras que têm esse nó como prefixo.

Método createNode:

Essa função cria um novo nó da Trie e inicializa seus membros. O array de ponteiros para os nós filhos é configurado para nullptr, indicando que inicialmente não há filhos para esse nó. As demais variáveis são inicializadas com valores padrão.

Método insertWord:

Essa função insere uma palavra na Trie. Ela percorre cada caractere da palavra e verifica se o nó correspondente já existe na Trie. Caso não exista, um novo nó é criado e vinculado ao nó atual. Em seguida, o contador wordCount é incrementado para indicar que mais uma palavra tem esse nó como prefixo. Por fim, o último nó correspondente ao último caractere da palavra é marcado como uma palavra completa, e suas informações de posição e comprimento são atualizadas.

Método searchPrefix:

Essa função busca um prefixo na Trie. Ela percorre cada caractere do prefixo e verifica se o nó correspondente existe na Trie. Caso algum nó não seja encontrado, a função retorna nullptr. Se todos os caracteres do prefixo forem encontrados, o último nó correspondente ao último caractere do prefixo é retornado.

Void printPrefix:

Essa função imprime informações sobre um prefixo na Trie. Ela utiliza a função searchPrefix() para obter o nó correspondente ao prefixo. Se o nó existir e for uma palavra completa, o número de palavras que têm esse prefixo é impresso, juntamente com as informações de posição e comprimento da palavra no dicionário. Caso contrário, é informado que o prefixo não existe na Trie.

Void parseDictionary:

Essa função lê o arquivo do dicionário e insere as palavras na Trie. Ela abre o arquivo especificado pelo parâmetro filename. Em seguida, lê cada linha do arquivo, procurando pelo par de colchetes[' e ']' que delimita uma palavra. Se a linha conter uma palavra válida, ela é extraída e inserida na Trie usando a função insertWord(). A variável position é atualizada para acompanhar a posição atual no arquivo. Por fim, o arquivo é fechado.

Void searchWord:

Essa função busca uma palavra na Trie. Ela percorre cada caractere da palavra, verificando se o nó correspondente existe. Se um nó não for encontrado, a busca é interrompida. O prefixo formado pelos caracteres encontrados é passado para a função printPrefix() para imprimir as informações correspondentes.

Int main:

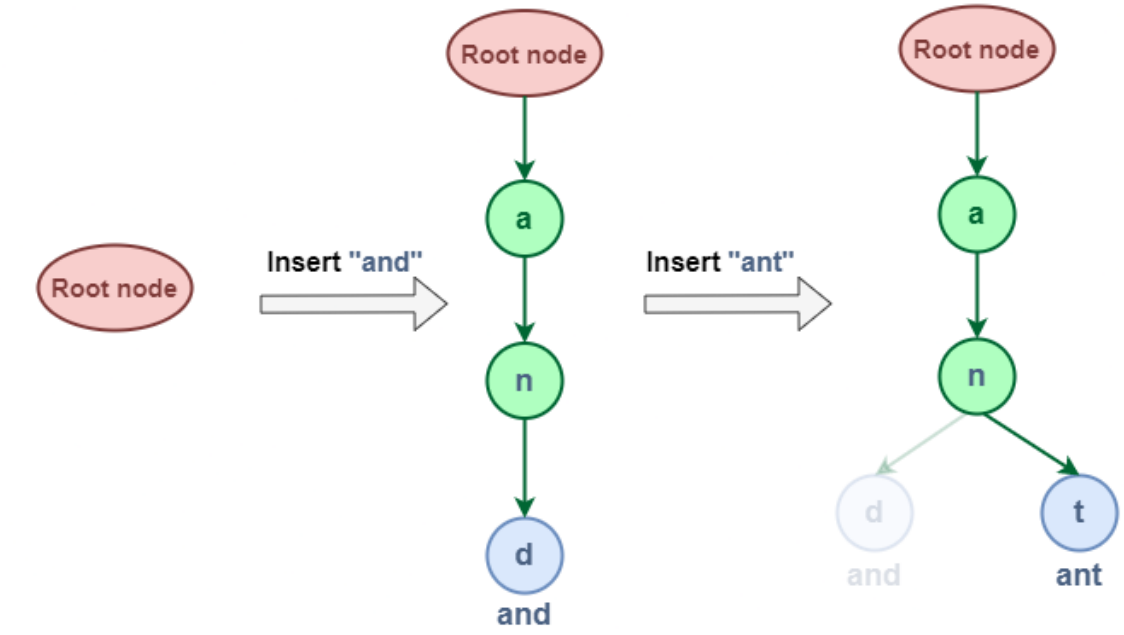
A função main() é a função principal do programa. Ela lê o nome do arquivo do dicionário a partir da entrada padrão. Em seguida, cria o nó raiz da Trie usando a função createNode() e preenche a Trie com as palavras do dicionário usando a função parseDictionary(). Depois disso, entra em um loop, onde lê palavras da entrada padrão até que a palavra "0" seja digitada. A cada palavra lida, chama a função searchWord() para buscar a palavra na Trie e imprimir as informações correspondentes. O programa termina quando a palavra "0" é digitada.

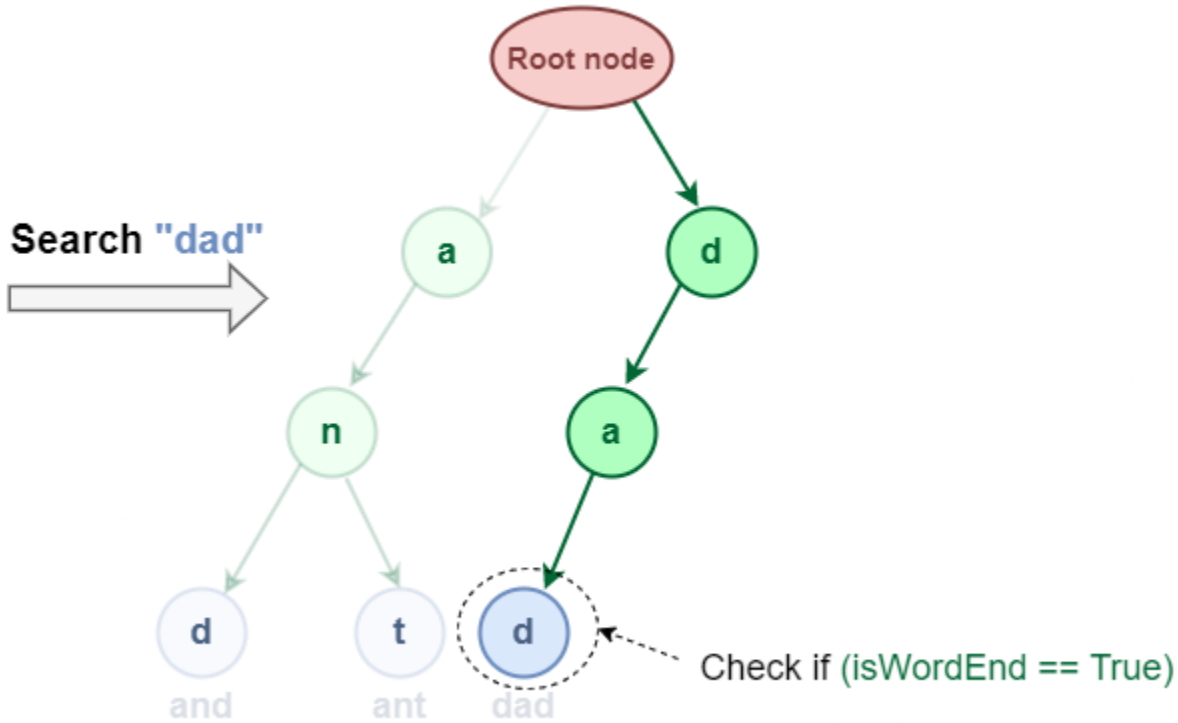
Conclusões :

A principal dificuldade que tive com esse trabalho foi entender como a Trie funciona e suas diferenças entre as outras árvores cobertas durante o semestre.
A sua estrutura e métodos implementados foram baseados em múltiplos exemplos que encontrei buscando aprender mais sobre a estrutura. Como principal referência utilizei o site https://www.geeksforgeeks.org/trie-insert-and-search/, lá todas as funções que a Trie deve apresentar são explicadas detalhadamente.

Figuras:

Essas figuras exemplificam o funcionamento das duas funções principais da Trie, buscar palavras e adicionar novas à estrutura:


Root node
Insert "and"
Root node
a
n
d
and
Insert "ant"
Root node
a
n
d
and
t
ant


Search "dad"
Root node
a
d
n
a
d
and
t
ant
d
dad
Check if (isWordEnd == True)