

# Laboratório 8: Busca Digital

*Entrega até domingo, 19/5, às 23:59h*

Em sala, discutimos a ideia de busca digital a partir de uma Árvore Trie. Nesse trabalho, iremos realizar a implementação das operações de inserção, remoção e “sugestão”, com o objetivo de adicionar e remover chaves em um certo conjunto e, dado um prefixo, conseguir sugerir formas de auto completção.

Neste trabalho, utilizaremos a implementação mais simples de Árvore Trie, onde cada nó possui um array contendo 26 posições (uma para cada letra do alfabeto), e um valor booleano informando se em um certo nó, forma um caminho para uma palavra. Para manipular as sequencias de caracteres, podemos utilizar as funções da biblioteca string.h (strlen, strcpy...). Lembre-se que toda string precisa terminar em '\0' para que as funções saibam quando a sequência de caracteres termina.

O programa de teste realiza a inserção de algumas palavras em `const char*`, onde o compilador adiciona um '\0' ao final de cada palavra. Depois, remove duas palavras do conjunto de chaves criado, e depois chama alguns métodos para imprimir todas as chaves adicionadas, e sugerir palavras a partir de um prefixo. A partir do teste implementado, faça as seguintes questões:

1. (3.0) Faça a implementação da função *insere*, criando os nós intermediários para representar cada dígito de uma chave. Lembre-se que o último nó precisa informar se o caminho percorrido forma uma palavra. Para transformar um dígito em um índice no array de nós, utilize o método *transformaparaindice*.
2. (4.0) Faça a implementação da função *retira*, que remove uma chave da árvore. Lembre-se que, a chave só será removida se, após percorrer todos os nós de acordo com os dígitos da chave, o nó resultante for uma palavra. Além disso, deve-se remover os nós ao longo do caminho que não pertencem a nenhuma outra chave (**imprima os dígitos removidos utilizando a função printf**). O arquivo **arvtrie.c** contém as respostas que devem ser mostradas no console caso a chave não for encontrada, caso ela tenha sido removida com sucesso, ou caso chegue em um nó que não é uma palavra. Obs: Durante a busca, você pode guardar um ponteiro para o último nó que for uma palavra, ou que possuir mais de um filho, para saber onde iniciar a remoção de nós sem precisar usar recursão.
3. (3.0) Por fim, termine a implementação da função *sugerir*, que recebe um prefixo, e mostra todas as chaves salvas na árvore que possuem tal prefixo. As sugestões só serão impressas caso todo o prefixo exista na árvore, caso contrário deve ser mostrada uma mensagem informando que o prefixo não foi encontrado. Obs: você pode usar a função *imprimechaves* como função auxiliar.

Faça upload dos arquivos **arvtrie.c** no EAD até dia 19 de maio, domingo, às 23:59h. Lembre-se de fazer a entrega mesmo que não tenha chegado ao final do exercício.