

Algoritmos e Estruturas de Dados 2

Trabalho 1

Prof. Ricardo Oliveira - 2024/1

Enunciado

O trabalho consiste na implementação de um Dicionário (des chaves inteiras e conteúdos *string*) na linguagem C, utilizando como estrutura base uma árvore **AVL** ou uma árvore **Rubro-Negra**, à sua escolha. A estrutura deve suportar as operações de inserção e de busca, além da obtenção de sucessor e antecessor, e do percurso *preorder*. Não é necessário implementar a operação de remoção.

Entrada

Seu programa deve ler comandos da entrada padrão. A entrada contém várias linhas, e cada linha contém uma operação a ser realizada. Cada linha pode ser de um dos seguintes formatos:

- **+ x c**: Insere um elemento com chave **x** e conteúdo **c** na árvore. A chave é um número inteiro e o conteúdo é uma *string* qualquer. Se já houver um elemento com chave **x** na árvore, a inserção não é realizada.
- **? x**: Busca um elemento de chave **x** na árvore.
- **> x**: Encontra o sucessor do elemento de chave **x** na árvore, caso tal elemento esteja presente;
- **< x**: Encontra o antecessor do elemento de chave **x** na árvore, caso tal elemento esteja presente;
- **#**: Realiza percurso *preorder* na árvore.

Saída

Seu programa deve produzir sua saída na saída padrão.

- Para cada comando de inserção cuja chave já existe na árvore, imprima uma linha com **elemento repetido**. Para cada inserção bem sucedida, imprima uma linha em branco;
- Para cada comando de busca, se o elemento não for encontrado, imprima uma linha com **nao encontrado**. Caso contrário, imprima uma linha com o conteúdo referente ao elemento encontrado;
- Para cada comando de sucessor, imprima uma linha com a chave e o conteúdo do sucessor encontrado, separados por um espaço. Se o elemento dado for o maior da árvore, imprima **maior**. Se o elemento dado não existe na árvore, imprima **chave invalida**;
- Para cada comando de antecessor, imprima uma linha com a chave e o conteúdo do antecessor encontrado, separados por um espaço. Se o elemento dado for o menor da árvore, imprima **menor**. Se o elemento dado não existe na árvore, imprima **chave invalida**;
- Para cada comando de percurso *preorder*, imprima uma linha com as todas as chaves da árvore, na ordem percorrida, separadas por um espaço. Não imprima um espaço ao final da lista. Se a árvore estiver vazia, imprima **arvore vazia**.

Exemplo de entrada	Exemplo de saída
#	arvore vazia
+ 42 e nessa loucura	
#	42
+ 50 de dizer que nao te quero	elemento repetido
+ 42 vou negando as aparencias	e nessa loucura
? 42	nao encontrado
? 23	
+ 23 disfarcando as evidencias	
? 23	disfarcando as evidencias
< 30	chave invalida
< 50	42 e nessa loucura
< 23	menor
> 30	chave invalida
> 50	maior
> 42	50 de dizer que nao te quero
#	42 23 50
+ 10 mas pra que viver fingindo	
+ 8 se eu nao posso enganar meu coracao	
#	42 10 8 23 50

Exemplo

Este é apenas um exemplo de entrada. Seu trabalho será corrigido com outros exemplos, diferentes e maiores.

Orientações

- O trabalho pode ser feito por equipes de *no máximo* 2 (dois) estudantes;
- Submeta, via *Moodle*, um pacote **zip** ou **tar.gz** contendo todo o código-fonte necessário para compilar e executar sua solução, além de um arquivo de texto (txt) onde conste:
 - O nome de todos os integrantes da equipe;
 - Toda informação que a equipe julgar relevante para a correção (como *bugs* conhecidos, detalhes de implementação, escolhas de projeto, etc.)
- Comente adequadamente seus códigos para facilitar a correção;
- **Atenção:** a correção será parcialmente automatizada, e a saída do programa será testada com outras entradas além das fornecidas como exemplo. *Siga **fielmente** o formato de entrada e de saída dado nos exemplos*, sob pena de grande redução da nota;
- **Atenção:** Haverá *nota extra* de até 10 pontos (o trabalho pode valer 110 pontos ao todo) de acordo com o desempenho de seu programa. O trabalho **funcionando corretamente** mais eficiente (isto é, com menor tempo de execução para as entradas grandes usadas na correção) receberá 10 pontos de nota extra. Os demais trabalhos **funcionando corretamente** receberão nota extra proporcional em relação ao trabalho mais eficiente. As entradas grandes não serão fornecidas!
- Certifique-se que seu programa compila e funciona antes de submetê-lo;
- O trabalho deve ser entregue até **9 de Maio de 2024, 23:59**, apenas via *Moodle*. Trabalhos entregues por outros meios ou fora do prazo não serão aceitos. É suficiente que o trabalho seja submetido por apenas um estudante da equipe;
- Trabalhos detectados como cópia/plágio (de colegas, da internet ou de ferramentas de IA), ou comprados, receberão **todos** a nota 0 (**ZERO**) e estarão sujeitos a abertura de Processo Administrativo Disciplinar Discente.