

sqtpm

[ggm]

[ajuda](#)[voltar](#)**Trabalho:** 25-PTCN

Data de abertura: 05/06/2025 17:00:00

Data limite para envio: 15/06/2025 23:59:00 ([aberto](#))

Número de arquivos a enviar: entre 1 e 3

Número máximo de envios: 30

Número de envios feitos por ggm: 0

Casos-de-teste abertos: [casos-de-teste.tgz](#)

Recursos: cpu 1 s, memória 32768 kb, stack 8192 kb

Enviar:

Linguagem:

C ▾

Arquivos:

Browse...

No files selected.

Enviar

Árvore PTCN

Macunaíma tem um grande conjunto de dados sobre os quais quer fazer buzilhões de buscas, inserções e remoções aleatorizadas.

Recentemente ele ficou sabendo das árvores AVL e achou uma boa idéia armazenar os dados em uma árvore parecida com a AVL porém mais relaxada, que foi batizada de PTCN.

A árvore PTCN respeita a propriedade de busca de uma árvore de busca binária tradicional (ABB). Na PTCN a diferença entre as alturas de duas subárvores irmãs deve estar no intervalo discreto $[-2, +2]$.

A inserção na PTCN é feita como na ABB, mas se a inserção fizer com que a árvore deixe de ser PTCN então a inserção é desfeita e a árvore permanece como estava antes da tentativa de inserção.

A remoção na PTCN é feita como na ABB, sendo que uma remoção de nó com dois filhos sempre usa o sucessor. Além disso, se a remoção fizer com que a árvore deixe de ser PTCN então ela é desfeita e a árvore permanece como estava antes da tentativa de remoção.

Escreva um programa em C que mantém uma árvore PTCN e realiza operações sobre ela.

Entrada

A entrada para o programa é uma sucessão de comandos, um por

sqtpm

[ggm]

[ajuda](#)

[voltar](#)

linha. Os comandos que seu programa deve interpretar são da seguinte forma:

- criar

Ao ler o comando criar o programa deve criar uma árvore PTCN vazia. Vamos chamá-la de T. Se já houver uma árvore sendo processada, todos os nós dela devem ser removidos.

- inserir k

Ao ler o comando inserir o programa deve inserir um novo nó com chave k e timestamp t em T. k é um número inteiro.

O timestamp é um inteiro que recebe valor zero quando T é criada e é incrementado de uma unidade depois de cada comando inserir e remover, seja ele bem-sucedido ou não.

Se já houver um nó com chave k em T então a árvore deve permanecer inalterada.

Se não houver memória para a operação inserir, o programa deve imprimir "memoria insuficiente" em uma linha e deve continuar a execução sem modificar T.

- remover k

Ao ler o comando remover o programa deve remover o nó com chave k de T. k é um inteiro.

Se não houver um nó com chave k em T então a árvore deve permanecer inalterada.

- buscar k

Ao ler o comando buscar, o programa deve buscar o registro com chave k em T e imprimir "chave k: inserida no tempo t" ou "nao ha chave k" em uma linha. k é um inteiro.

- imprimir

Ao ler esse comando o programa deve imprimir as chaves na ordem em que forem visitadas por um percurso em profundidade em-ordem. O formato está exemplificado abaixo. Se a árvore estiver vazia então o programa deve imprimir "arvore vazia" em uma linha.

- info

Ao ler esse comando o programa deve imprimir o número de nós da árvore, o número de folhas na árvore e a altura da

sqtpm

[ggm]

[ajuda](#)
[voltar](#)

arvore, no formato "nos: x, folhas: z, altura: h". Se a árvore for vazia, o comando deve considerar a altura igual a zero.

- terminar

Ao ler esse comando o programa deve desalocar a árvore e todos os nós dela e terminar.

Exemplos de entrada e saída

Entrada	Saída
criar inserir 70 inserir 50 inserir 55 inserir 30 inserir 20 inserir 65 inserir 90 inserir 80 imprimir info buscar 55 buscar 10 buscar 70 remover 90 remover 80 remover 65 imprimir remover 50 remover 30 remover 70 remover 55 remover 20 imprimir terminar	em-ordem: 30 50 55 70 80 90 nos: 6, folhas: 3, altura: 2 chave 55: inserida no tempo 2 nao ha chave 10 chave 70: inserida no tempo 0 em-ordem: 30 50 55 70 arvore vazia

Avaliação

- Trabalhos com algoritmos de busca, inserção e remoção com tempo $O(\log n)$ para árvores com n nós terão bônus de 30% na nota deste trabalho.

Requisitos adicionais

- Seu programa deve liberar toda a memória que alocar dinamicamente antes do término da execução.
- Não pode haver qualquer variável global. Uma variável é global se estiver declarada fora de alguma função (variáveis declaradas dentro da main não são globais, são locais à

função main).

sqtpm

[ggm]

[ajuda](#)

[voltar](#)

Sugestões

- Comece fazendo um programa que apenas lê a entrada e imprime. Depois adicione as demais funcionalidades.

Sobre organização do código e comentários

- Faça um programa organizado, bem indentado e que seja fácil de ler.
 - Adicione comentários que vão ser úteis para entender o programa se você for relê-lo daqui a alguns anos: comentar cada linha vai ser redundante; documentar blocos de código e a estratégia usada na solução vai ser muito útil.
-