Aula 08-05-2024 - Árvore Binária

Inserção

Adicionar elementos repetidos

- Caso queira, deve ser tomada uma decisão: o elemento repetido vai ficar na esquerda ou direita?

Inserção com retorno de referencia

- A ideia é retornar o endereço do nó acrescentado do elemento previamente pesquisado
- Caminha na árvore com dois ponteiros (o i vai caminhando até chegar em null, mas o que impede o código de dar problema é que o outro ponteiro está em seu pai, ou seja, ele será retornado)

Análise de complexidade

- Melhor caso: teta(1) - Inserindo na raíz
- Pior caso: teta(n) - Ordem crescente ou decrescente
- Caso Médio: teta(log(n))

Pesquisa e Caminhamento

Funcionamento Básico de Pesquisa

1 - Se a raíz estiver vazia, retornar **pesquisa negativa**

2 - Senão, se o elemento procurado for **igual** ao da raíz, retornar uma **pesquisa positiva**

3 - Se o elemento for menor que a raíz, comparar com o da esquerda

4 - Se o elemento for maior que a raíz, comparar com o da direita

Análise de Complexidade da Pesquisa

- Melhor caso: teta(1) - na raiz
- Pior caso: teta(n) - quando inserimos em ordem, crescente ou decrescente
- Caso medio teta(log(n)) - árvore balanceada

Caminhamento

> ⚠ Passar por todos os nós da árvore **CUSTO teta(n) visitas**

Exercicio

Insira os elementos em uma árvore e em seguida faça o caminhar central. Justifique por que esse algoritmo é conhecido como treesort

Fazer Exercício 8 do PDF

Inserir em C

```
void inserirRec(int x, No** i){
    if(*i == NULL){
        i* = novoNo(x);
    } else if(x<(*i)->elemento){
        inserirRec(x, &((*i)->esq);
    } else if(x > (*i)->elemento){
        inserirRec(x,&((*i)->dir);
    } else{
        errx(1,"Erro ao inserir");
    }
}
```

EXERCICIOS PARA SEGUNDA FEIRA

Pegar último slide de árvore (QUESTOES Q TEM 100% DE CHANCE DE CAIR NA PROVA)