unipac.br
Barbacena

Bacharelado em Ciência da Computação

Estruturas de Dados

Material de Apoio

Parte XVII – *Pesquisa em Árvores*

Prof. Nairon Neri Silva
naironsilva@unipac.br

2º sem / 2021

BUSCA

Diversas aplicações precisam buscar um determinado valor em um conjunto de dados. Essa busca deve ser feita da forma mais eficiente possível.

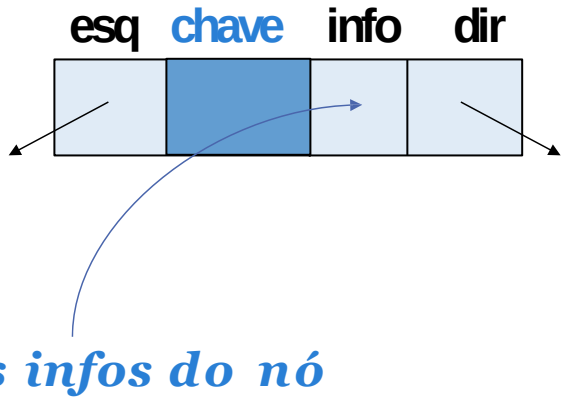
Árvores binárias possibilitam buscas com eficiência.

Exemplo: buscar dados de uma pessoa que possui um determinado CPF. Dados das pessoas são armazenados numa árvore binária de busca.

CPF funciona como “chave”, pois é único para cada pessoa (não existem duas pessoas com o mesmo CPF)

ÁRVORES BINÁRIAS DE BUSCA

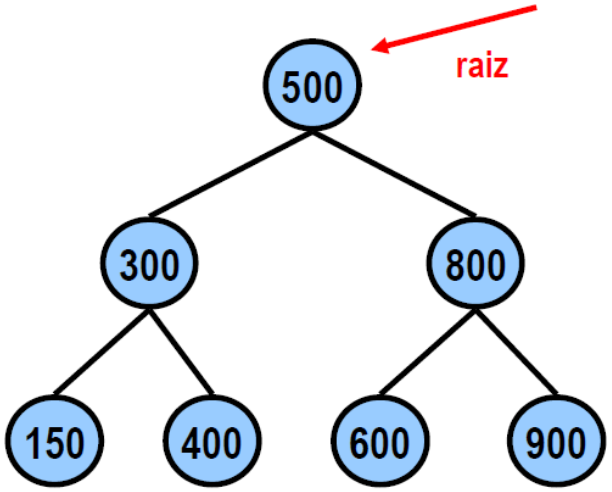
Apresentam uma relação de ordem entre os nós. A ordem é definida pela chave



ÁRVORES BINÁRIAS DE BUSCA

Uma árvore binária T é uma árvore binária de busca se:

- Chaves da subárvore esquerda de T são menores do que chave da raiz de T; e
- Chaves da subárvore da direita de T são maiores do que a chave da raiz de T; e
- Subárvores da esquerda e da direita de T são árvores binárias de busca

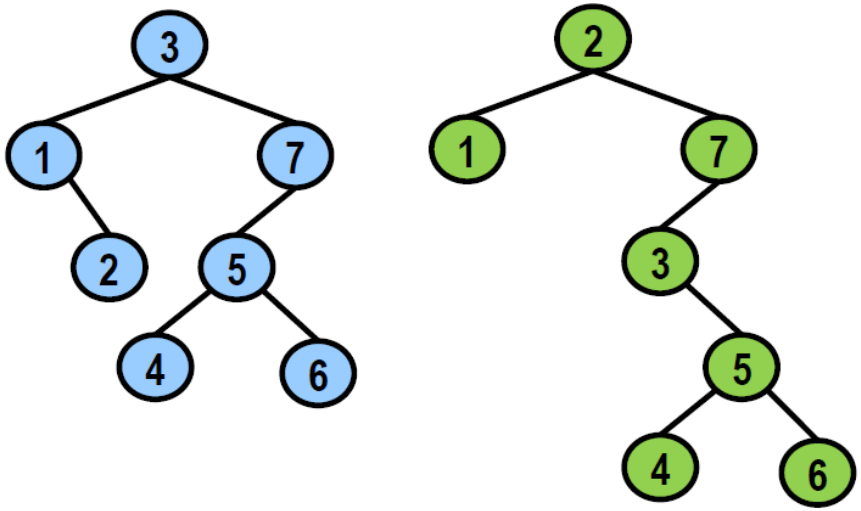


ÁRVORES BINÁRIAS DE BUSCA

Para um mesmo conjunto de chaves, existem várias árvores binárias de busca possíveis

Exemplos para o conjunto de chaves:

{1, 2, 3, 4, 5, 6, 7}



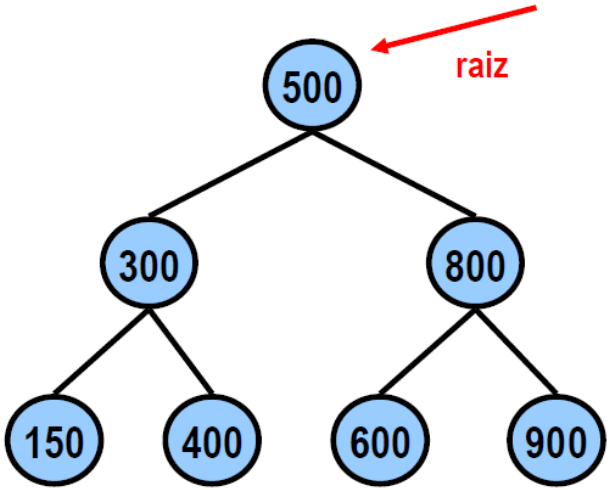
BUSCA POR NÓ COM CHAVE X

Em qualquer nó:

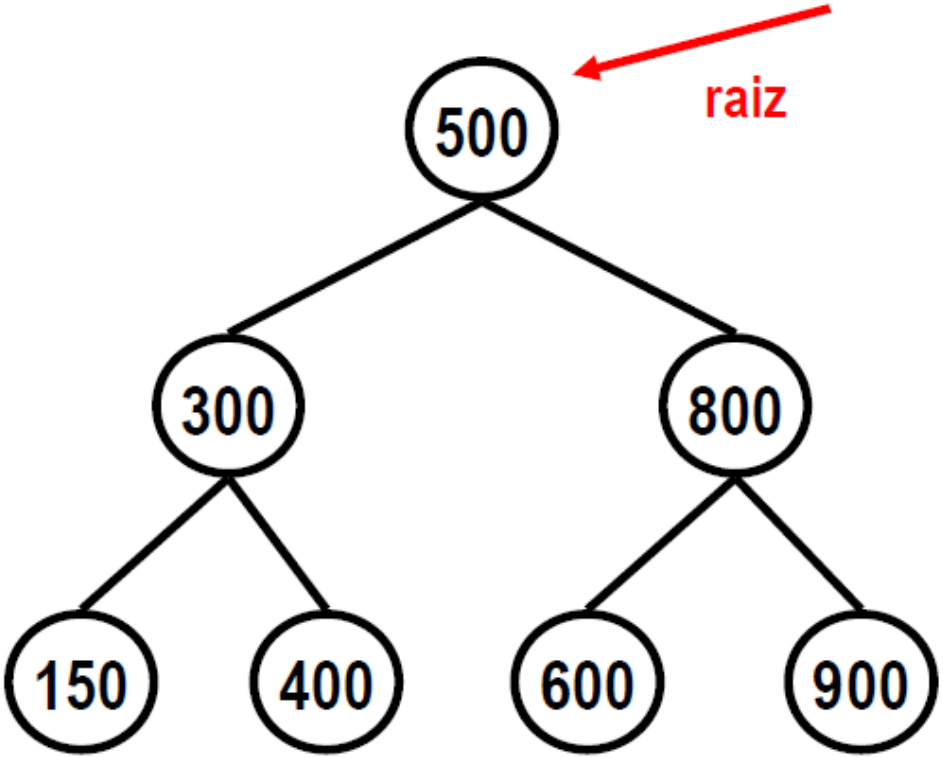
X = Chave

X > Chave

X < Chave



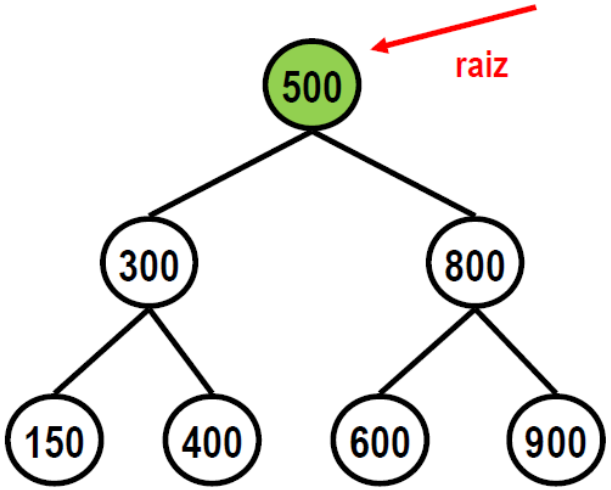
EXEMPLO: BUSCAR POR 600



EXEMPLO: BUSCAR POR 600

600 **>** 500

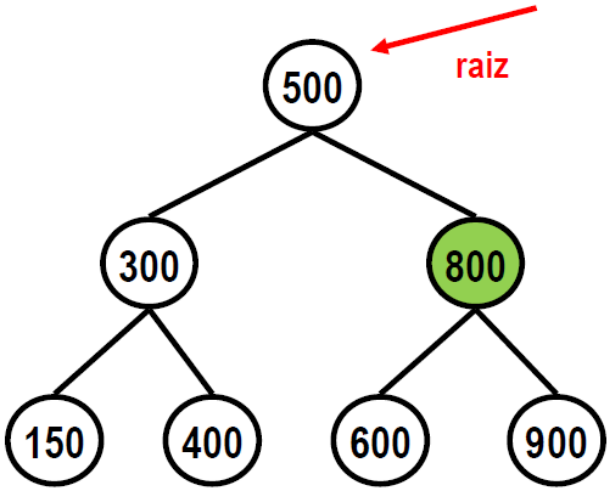
Ir para direita



EXEMPLO: BUSCAR POR 600

600 < 800

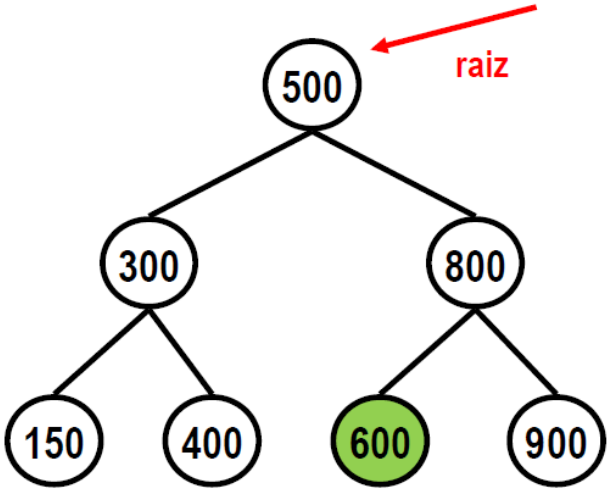
Ir para a esquerda



EXEMPLO: BUSCAR POR 600

600 = 600

Achou



EXEMPLO: BUSCAR POR 200

200 < 500

Ir para esquerda

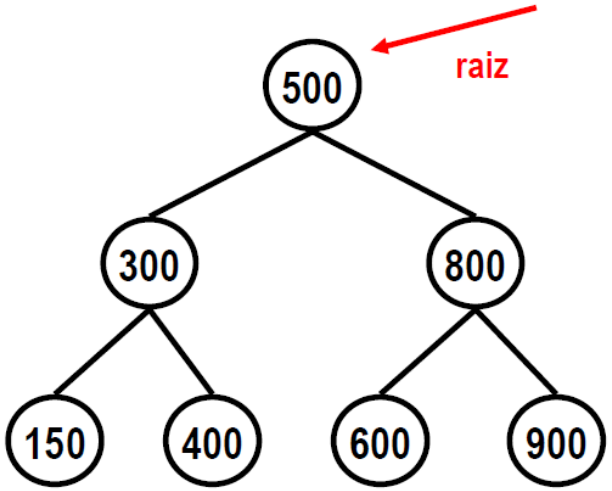
200 < 300

Ir para esquerda

200 > 150

Ir para a direita

NULL: chave não encontrada

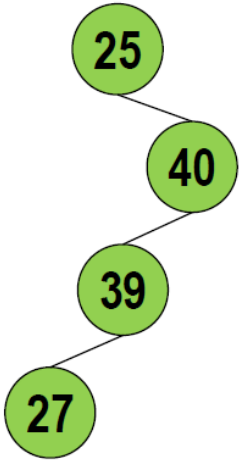


PROBLEMA

A ordem em que as chaves são inseridas numa árvore de busca binária pode fazer com que uma árvore se deteriore, ficando com altura muito grande.

Exemplo:

25 40 39 27



CRIAÇÃO DE ÁRVORE BINÁRIA DE BUSCA MAIS BALANCEADA POSSÍVEL

Sabendo disso, é possível reordenar as chaves de entrada de forma a obter uma árvore o mais balanceada possível.

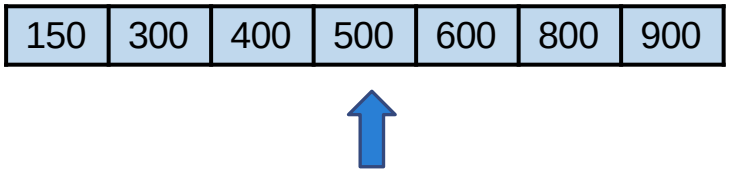
Algoritmo:

- Seja v um vetor ORDENADO contendo as chaves a serem inseridas
- Inserir a chave do meio
- Chamar recursivamente para os dois pedaços que sobraram

CRIAÇÃO DE ÁRVORE BINÁRIA DE BUSCA MAIS BALANCEADA POSSÍVEL

Algoritmo:

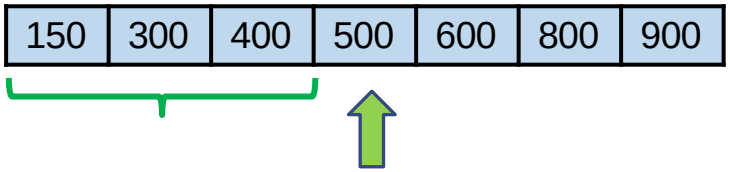
- Seja v um vetor ORDENADO contendo as chaves a serem inseridas
- Inserir a chave do meio
- Chamar recursivamente para os dois pedaços que sobraram (esquerda e direita)



CRIAÇÃO DE ÁRVORE BINÁRIA DE BUSCA MAIS BALANCEADA POSSÍVEL

Algoritmo:

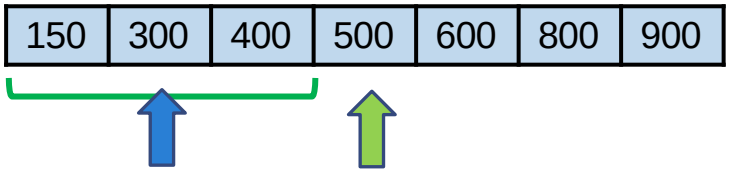
- Seja v um vetor ORDENADO contendo as chaves a serem inseridas
- Inserir a chave do meio
- Chamar recursivamente para os dois pedaços que sobraram (esquerda e direita)

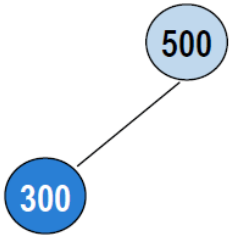


CRIAÇÃO DE ÁRVORE BINÁRIA DE BUSCA MAIS BALANCEADA POSSÍVEL

Algoritmo:

- Seja v um vetor ORDENADO contendo as chaves a serem inseridas
- Inserir a chave do meio
- Chamar recursivamente para os dois pedaços que sobraram (esquerda e direita)

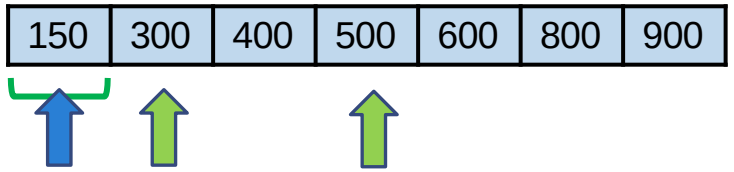


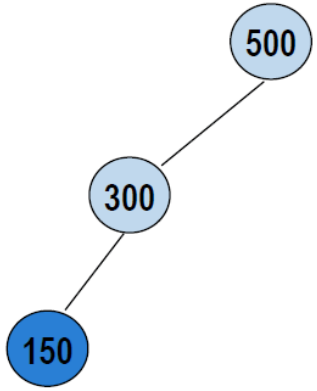


CRIAÇÃO DE ÁRVORE BINÁRIA DE BUSCA MAIS BALANCEADA POSSÍVEL

Algoritmo:

- Seja v um vetor ORDENADO contendo as chaves a serem inseridas
- Inserir a chave do meio
- Chamar recursivamente para os dois pedaços que sobraram (esquerda e direita)

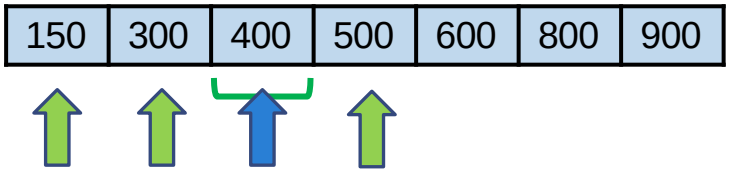


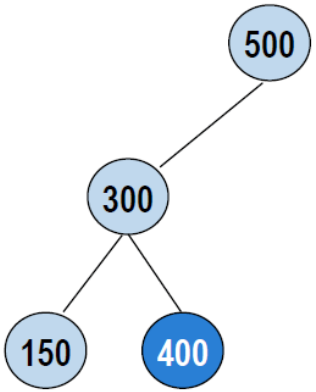


CRIAÇÃO DE ÁRVORE BINÁRIA DE BUSCA MAIS BALANCEADA POSSÍVEL

Algoritmo:

- Seja v um vetor ORDENADO contendo as chaves a serem inseridas
- Inserir a chave do meio
- Chamar recursivamente para os dois pedaços que sobraram (esquerda e direita)

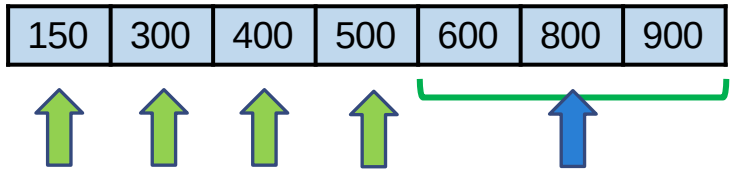


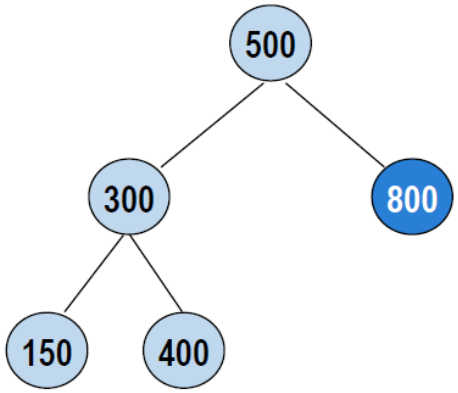


CRIAÇÃO DE ÁRVORE BINÁRIA DE BUSCA MAIS BALANCEADA POSSÍVEL

Algoritmo:

- Seja v um vetor ORDENADO contendo as chaves a serem inseridas
- Inserir a chave do meio
- Chamar recursivamente para os dois pedaços que sobraram (esquerda e direita)

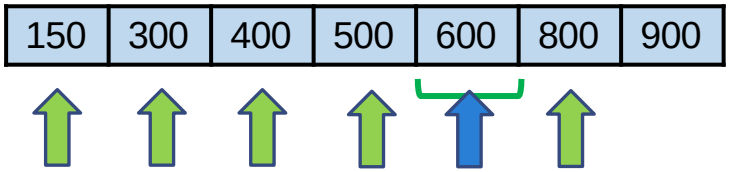


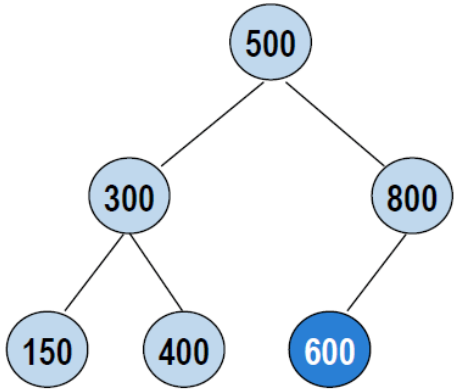


CRIAÇÃO DE ÁRVORE BINÁRIA DE BUSCA MAIS BALANCEADA POSSÍVEL

Algoritmo:

- Seja v um vetor ORDENADO contendo as chaves a serem inseridas
- Inserir a chave do meio
- Chamar recursivamente para os dois pedaços que sobraram (esquerda e direita)

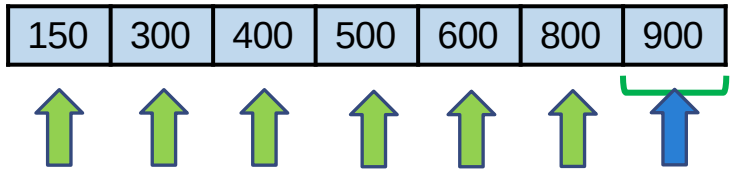


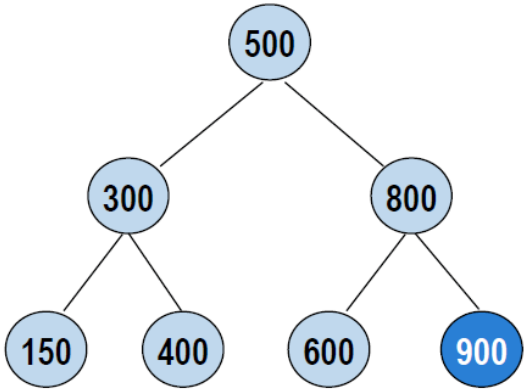


CRIAÇÃO DE ÁRVORE BINÁRIA DE BUSCA MAIS BALANCEADA POSSÍVEL

Algoritmo:

- Seja v um vetor ORDENADO contendo as chaves a serem inseridas
- Inserir a chave do meio
- Chamar recursivamente para os dois pedaços que sobraram (esquerda e direita)

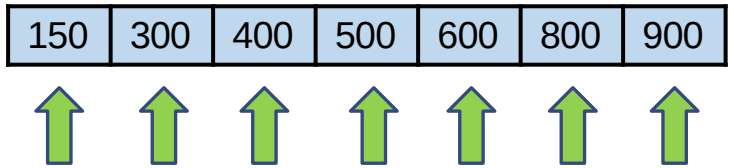


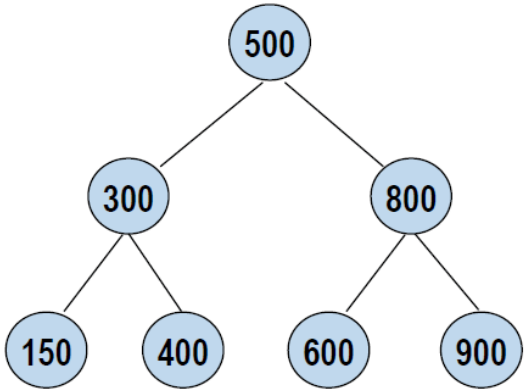


CRIAÇÃO DE ÁRVORE BINÁRIA DE BUSCA MAIS BALANCEADA POSSÍVEL

Algoritmo:

- Seja v um vetor ORDENADO contendo as chaves a serem inseridas
- Inserir a chave do meio
- Chamar recursivamente para os dois pedaços que sobraram (esquerda e direita)





IMPLEMENTAÇÃO DA PESQUISA

```
/* Recebe k e uma árvore de pesquisa r. Devolve o nó cuja
   chave é k ou devolve NULL se tal nó não existe. */

no Busca (arvore r, int k)
{
  if (r == NULL || r->chave == k)
      return r;
  if (r->chave >  k)
     return Busca(r->esq, k);
  else
    return Busca(r->dir, k);
}
```

Referências

- Material baseado nos slides de Renata Galante, UFRGS
- Slides da Professora Vanessa Braganholo - Estruturas de Dados e Seus Algoritmos - UFF
- Szwarcfiter, J.; Markezon, L. Estruturas de Dados e seus Algoritmos, 3a. ed..