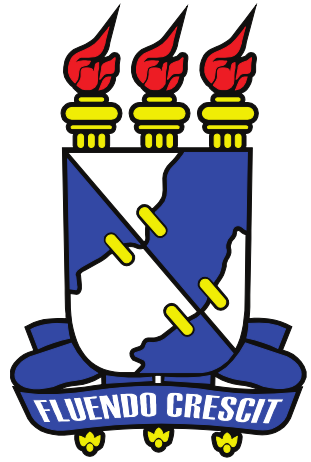


Estrutura de Dados I

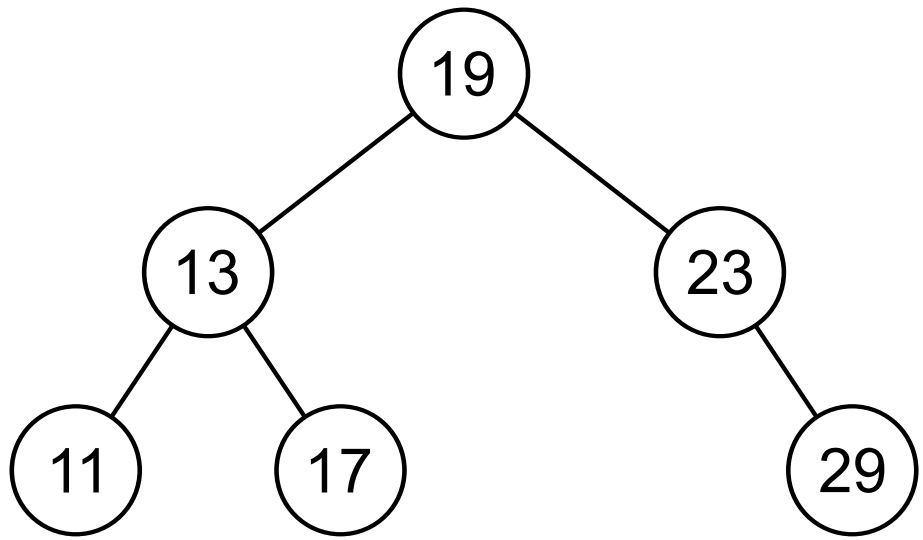
Busca em árvores binárias

Bruno Prado

Departamento de Computação / UFS

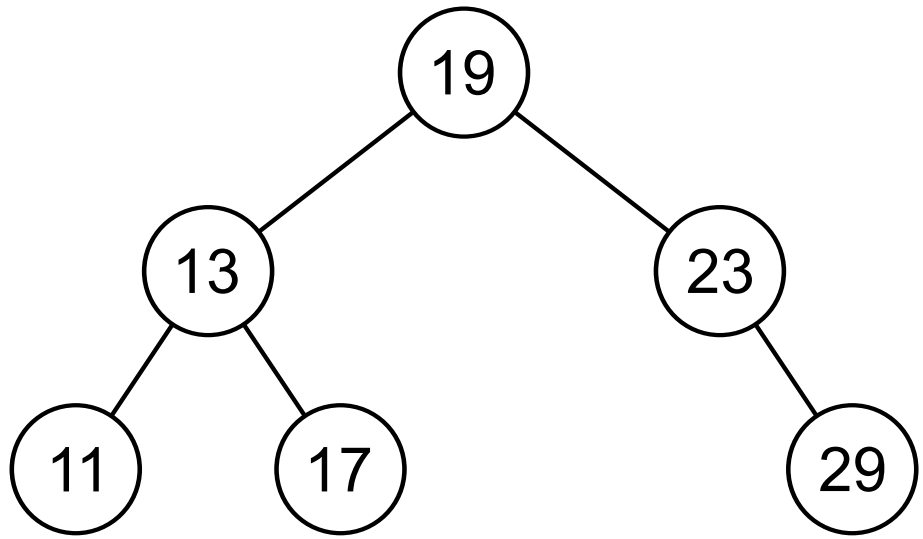
Introdução

- O que é uma árvore binária de busca?
  - É uma árvore enraizada onde cada nó possui uma chave associada para realização da busca
    - Esquerda: menores ou iguais
    - Direita: maiores ou iguais



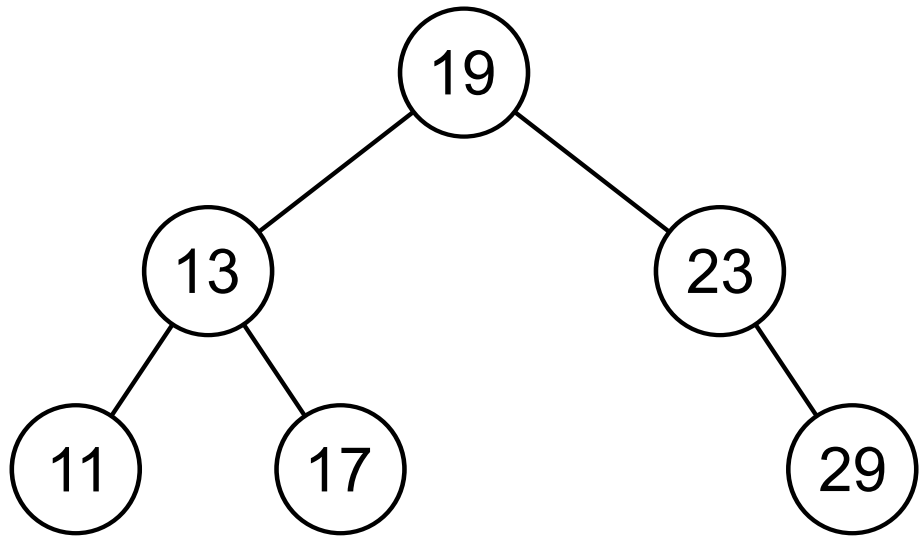
Introdução

- Propriedades
  - A realização do percurso em ordem em uma árvore binária fornece a sequência ordenada dos elementos



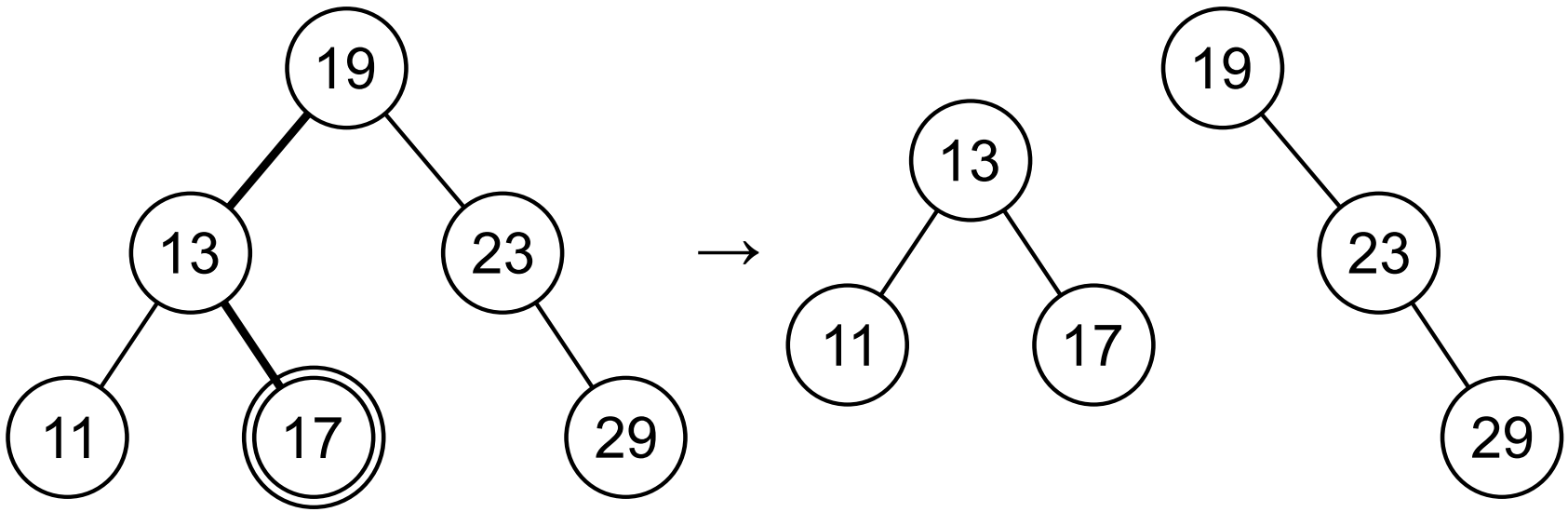
Introdução

- Propriedades
  - Para encontrar os nós mínimo e máximo, basta realizar um percurso até cada extremidade da árvore



Introdução

- Propriedades
  - Particionamento por chave k da árvore em duas subárvores com elementos menores e maiores que k

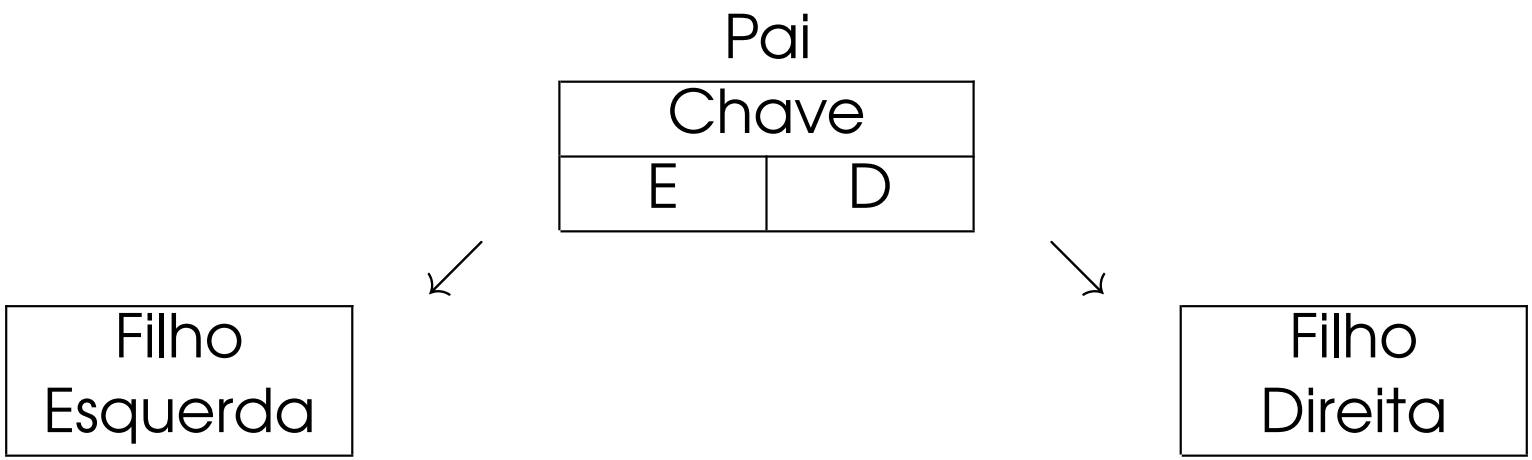


Busca em Árvores Binárias

- Operações básicas
  - Busca
  - Inserção
  - Remoção

Busca em Árvores Binárias

- Implementação em C
  - Estruturas e ponteiros



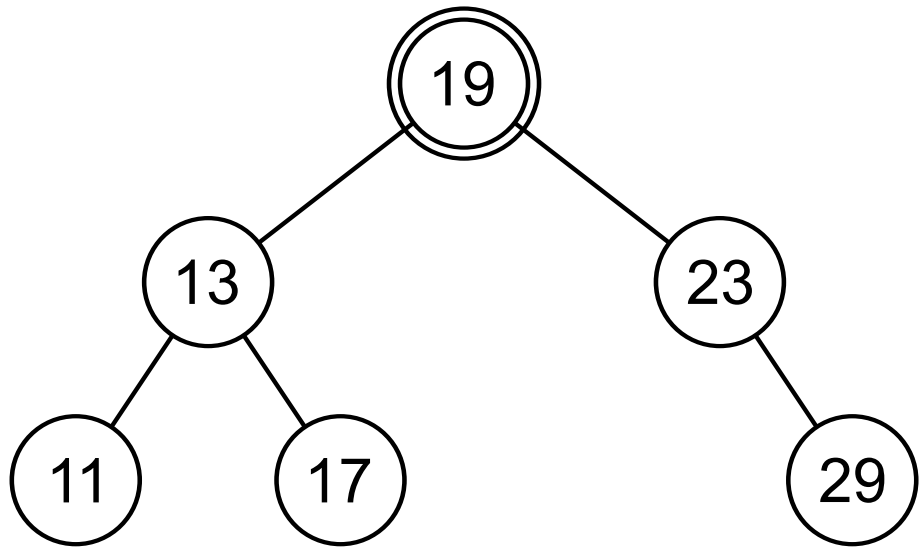
Busca em Árvores Binárias

- Implementação em C
  - Estruturas e ponteiros

```
// Representação da árvore binária
typedef struct no {
        // Chave do nó
        int chave;
        // Filho da esquerda
        struct no* E;
        // Filho da direita
        struct no* D;
} no;
```

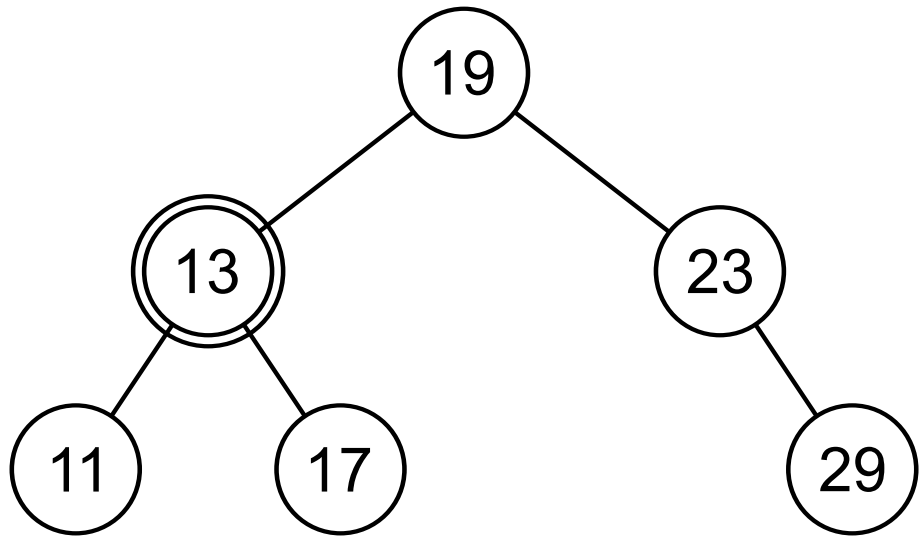
Busca em Árvores Binárias

- Operação de busca
  - Parâmetro de chave: 17
  - A busca tem início pelo elemento raiz da árvore, comparando o valor de sua chave com o parâmetro de busca



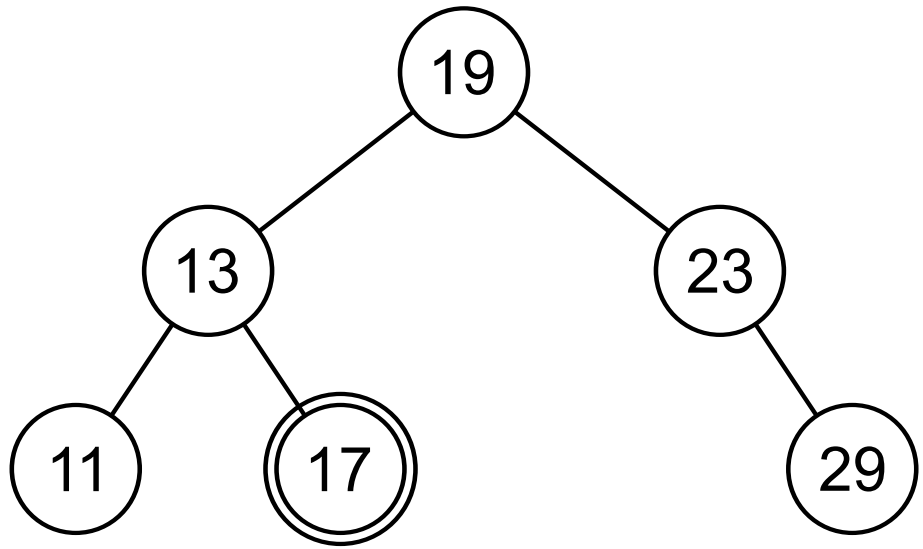
Busca em Árvores Binárias

- Operação de busca
  - Parâmetro de chave: 17
  - Como o resultado da comparação indica que o valor é maior do que o procurado, a busca é aplicada na subárvore esquerda



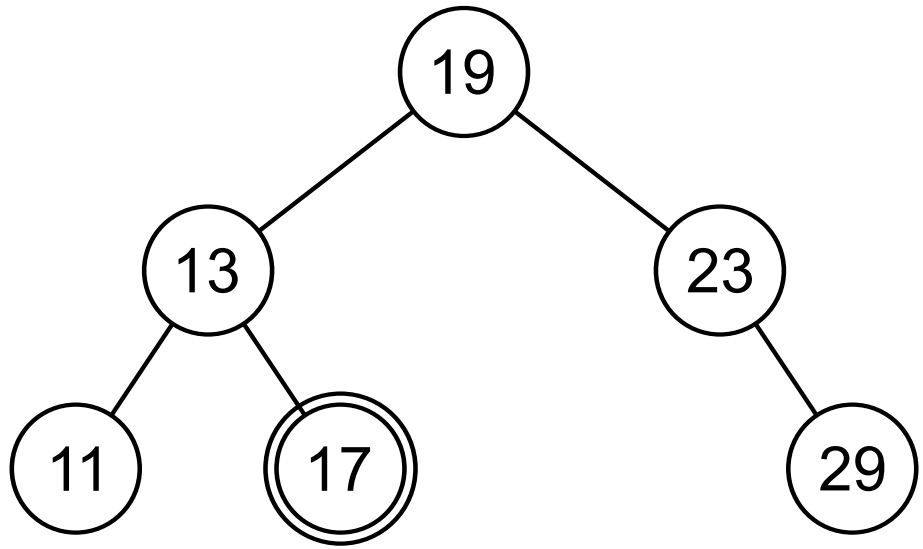
Busca em Árvores Binárias

- Operação de busca
  - Parâmetro de chave: 17
  - Como a comparação da chave do nó da subárvore indica que o parâmetro procurado é maior, a busca é aplicada na subárvore direita



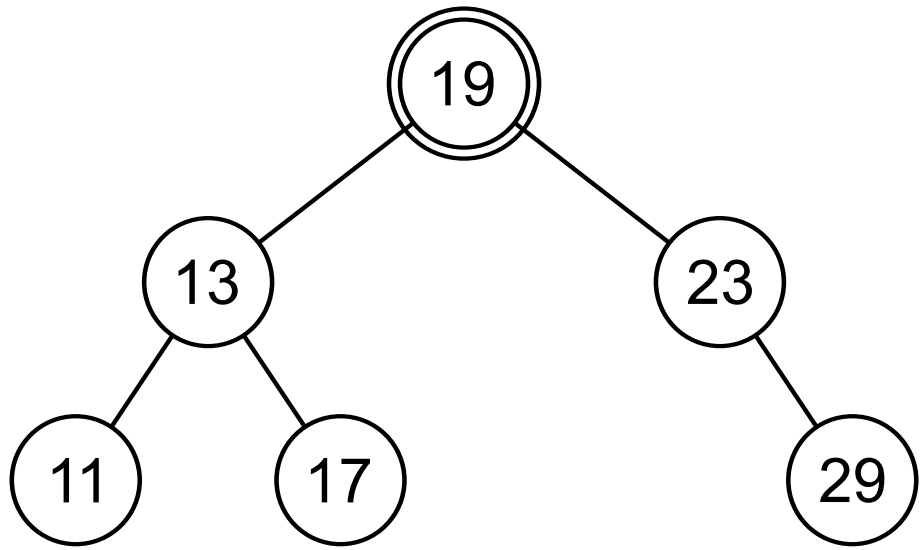
Busca em Árvores Binárias

- Operação de busca
  - Parâmetro de chave: 17
  - A chave do nó é igual ao parâmetro de busca e sua referência é retornada



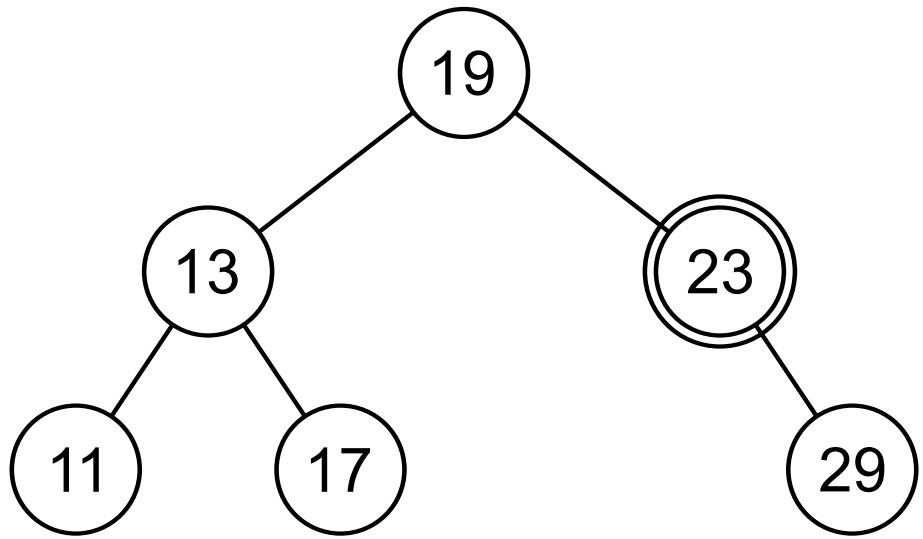
Busca em Árvores Binárias

- Operação de inserção
  - Parâmetro de chave: 21
  - É realizada uma operação de busca utilizando a chave do elemento que será inserido até encontrar uma referência nula



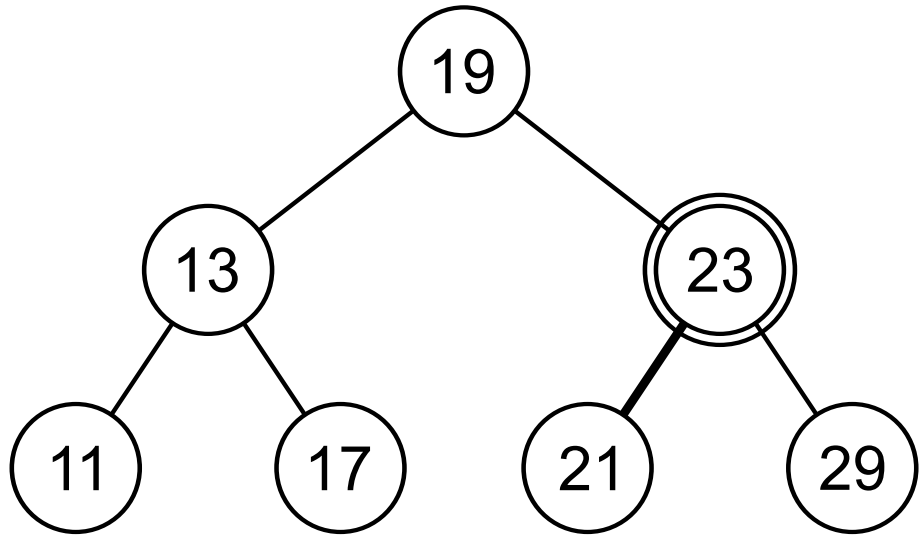
Busca em Árvores Binárias

- Operação de inserção
  - Parâmetro de chave: 21
  - É realizada uma operação de busca utilizando a chave do elemento que será inserido até encontrar uma referência nula



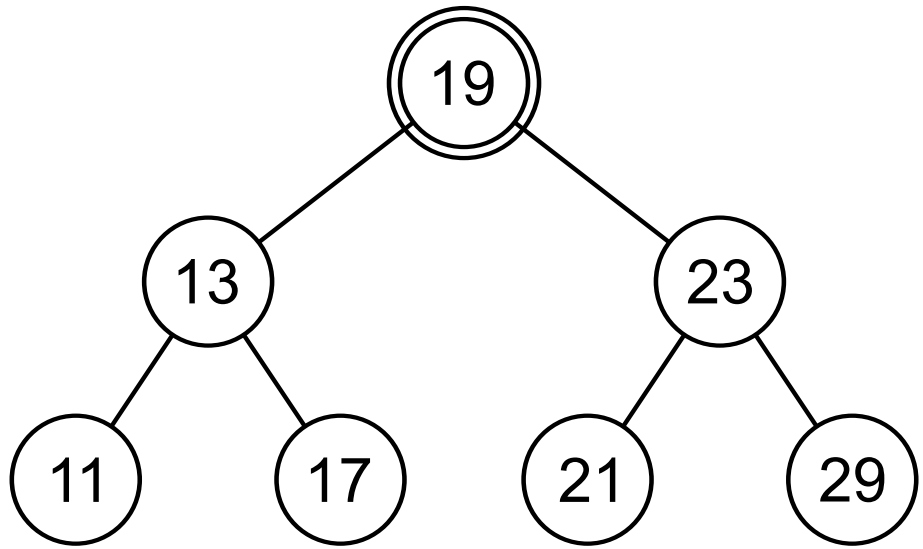
Busca em Árvores Binárias

- Operação de inserção
  - Parâmetro de chave: 21
  - A subárvore esquerda do nó 23 é nula, é feita a alocação do nó para inserção na árvore



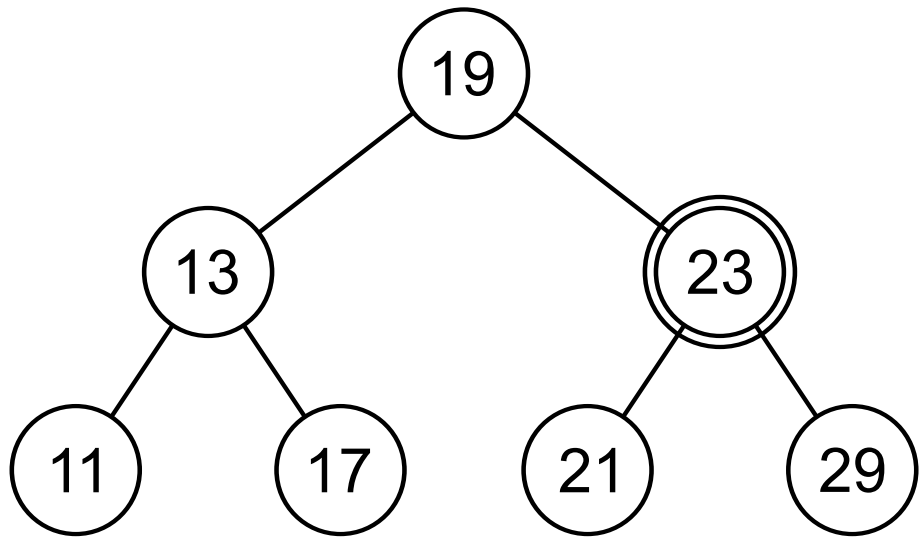
Busca em Árvores Binárias

- Operação de remoção
  - Caso 1: o nó removido é uma folha
    - Parâmetro de chave: 29
    - É feita a busca pela chave do elemento



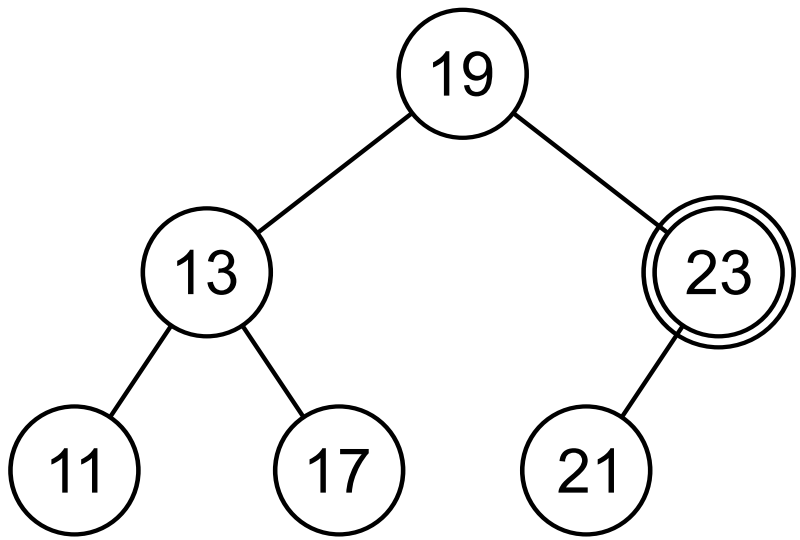
Busca em Árvores Binárias

- Operação de remoção
  - Caso 1: o nó removido é uma folha
    - Parâmetro de chave: 29
    - É feita a busca pela chave do elemento



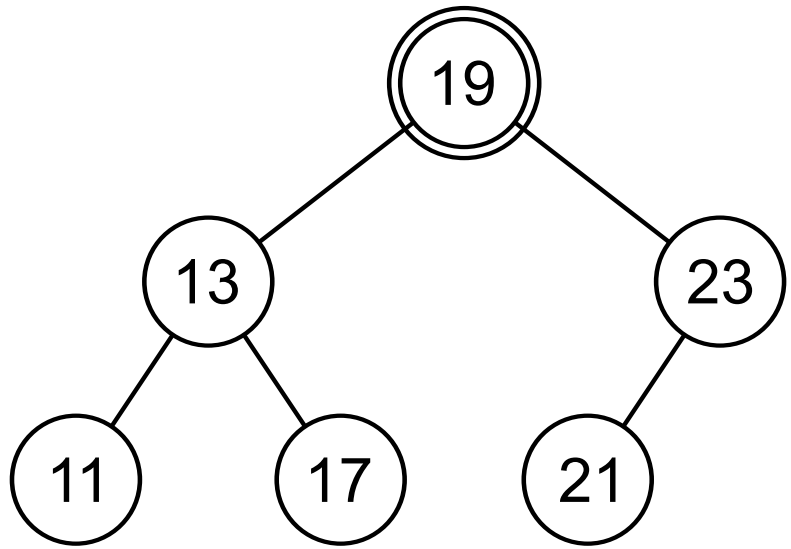
Busca em Árvores Binárias

- Operação de remoção
  - Caso 1: o nó removido é uma folha
    - Parâmetro de chave: 29
    - É feita a desalocação do elemento e sua referência é anulada para remoção da árvore



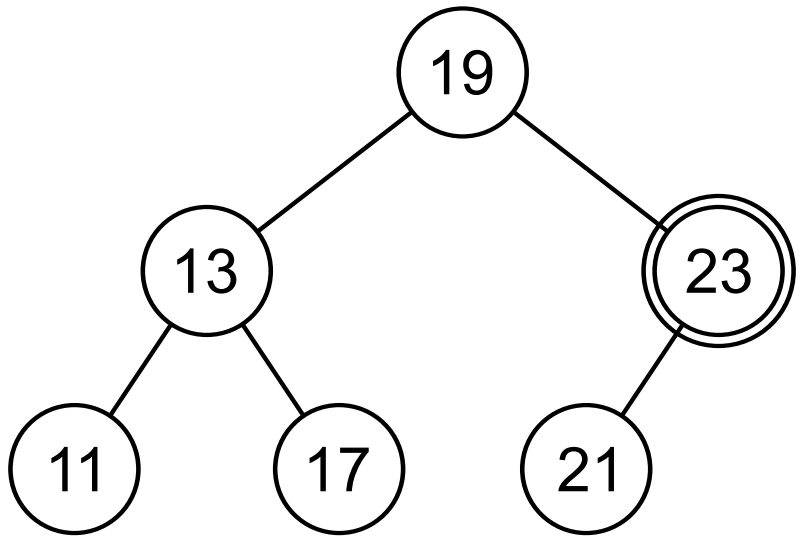
Busca em Árvores Binárias

- Operação de remoção
  - Caso 2: o nó removido possui uma subárvore
    - Parâmetro de chave: 23
    - É feita a busca pela chave do elemento



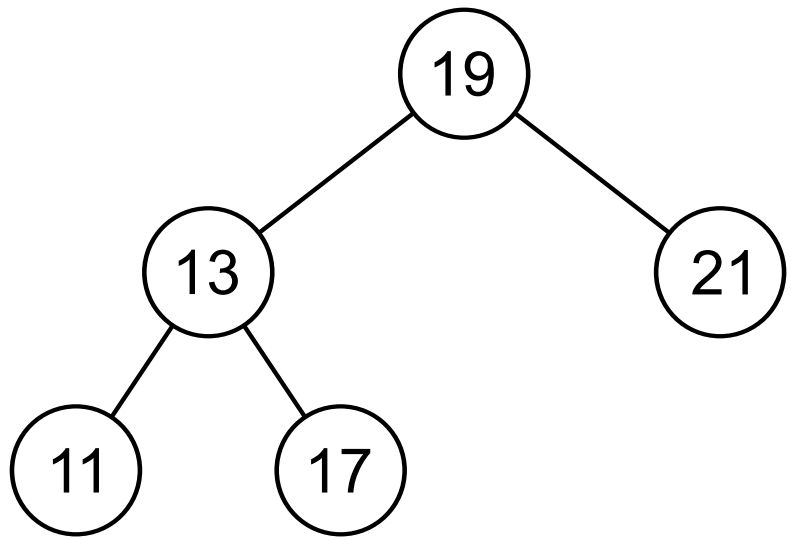
Busca em Árvores Binárias

- Operação de remoção
  - Caso 2: o nó removido possui uma subárvore
    - Parâmetro de chave: 23
    - É feita a busca pela chave do elemento



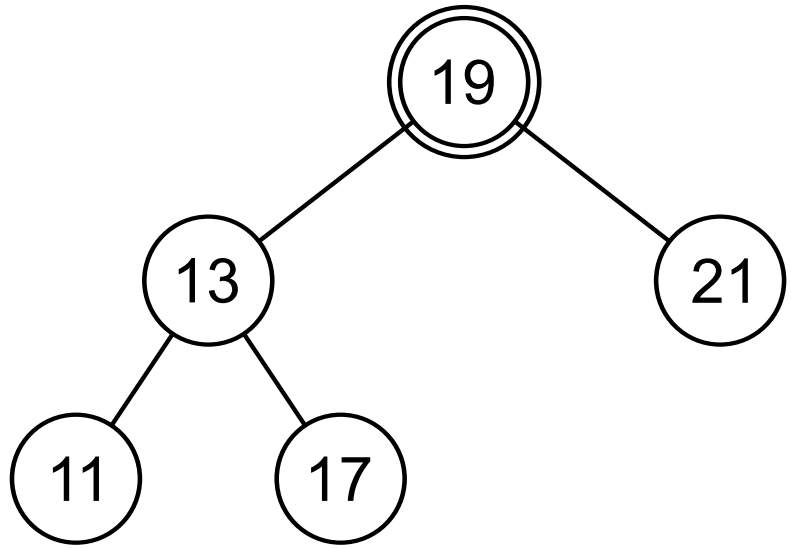
Busca em Árvores Binárias

- Operação de remoção
  - Caso 2: o nó removido possui uma subárvore
    - Parâmetro de chave: 23
    - O elemento é removido e a referência do nó pai é atualizada para referenciar a subárvore



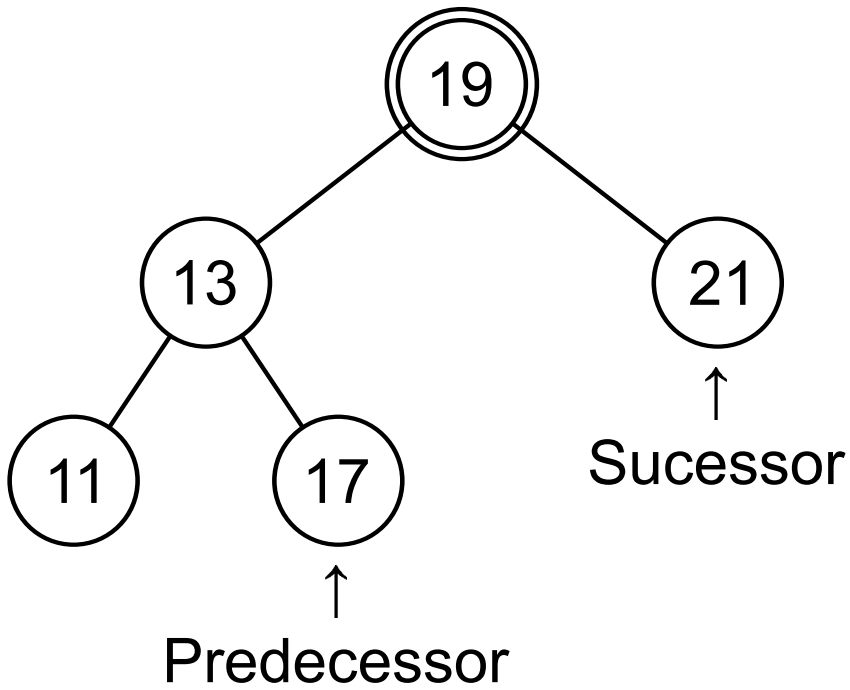
Busca em Árvores Binárias

- Operação de remoção
  - Caso 3: o nó removido possui duas subárvores
    - Parâmetro de chave: 19
    - É feita a busca pela chave do elemento



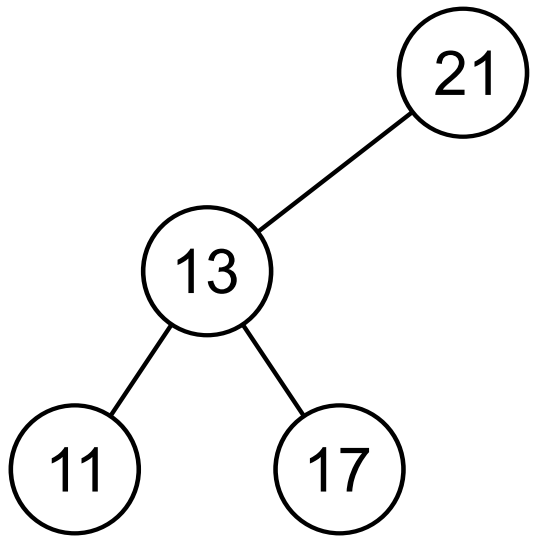
Busca em Árvores Binárias

- Operação de remoção
  - Caso 3: o nó removido possui duas subárvores
    - Parâmetro de chave: 19
    - O elemento raiz é substituído pelo seu predecessor ou sucessor que será removido após a substituição



Busca em Árvores Binárias

- Operação de remoção
  - Caso 3: o nó removido possui duas subárvores
    - Parâmetro de chave: 19
    - É feita a remoção do elemento de chave 21 e o ajuste dos ponteiros da árvore



Busca em Árvores Binárias

- Análise de complexidade
  - Espaço $\Theta(n)$
  - Tempo $\Omega(1)$ e $O(n)$

Uma árvore binária de altura h

$$\downarrow$$

$$\log_2 n \leq h \leq n$$

Exemplo

- Construa uma árvore binária de busca
  - Insira os elementos com chaves 13, 2, 34, 11, 7, 43 e 9
  - Realize a remoção dos elementos de chave 11 e 9
  - Explique as situações de melhor e de pior caso para as operações realizadas neste tipo de árvore

Exercício

- A empresa de tecnologia Poxim Tech está desenvolvendo um sistema de armazenamento de arquivos baseado em árvore binária
  - O formato de nome dos arquivos é definido por uma cadeia com 1 até 50 caracteres, composta somente por letras, números e os símbolos '_' e '.'
  - Cada arquivo possui também informações de permissão de acesso para somente leitura (ro) e escrita e leitura (rw), além do tamanho em bytes
  - Caso um nome de arquivo repetido seja inserido, é feita a substituição das informações desde que o arquivo permita a escrita (rw)

Exercício

- Formato de arquivo de entrada
  - [#*Número de arquivos*]
  - [$Nome_1$] [$Tipo_1$] [$Tamanho_1$]
  - ⋮
  - [$Nome_n$] [$Tipo_n$] [$Tamanho_n$]

```
5
lista_ed1.c rw 1123
senhas.txt ro 144
foto.jpg rw 8374719
documento.doc rw 64732
lista_ed1.c ro 1
```

Exercício

- Formato de arquivo de saída
  - Percurso em ordem (EPD), pré-ordem (PED) e pós-ordem (EDP)

```
EPD:
3 documento.doc rw 64732 bytes
2 foto.jpg rw 8374719 bytes
4 lista_ed1.c ro 1 byte
1 senhas.txt ro 144 bytes
PED:
4 lista_ed1.c ro 1 byte
2 foto.jpg rw 8374719 bytes
3 documento.doc rw 64732 bytes
1 senhas.txt ro 144 bytes
EDP:
3 documento.doc rw 64732 bytes
2 foto.jpg rw 8374719 bytes
1 senhas.txt ro 144 bytes
4 lista_ed1.c ro 1 byte
```