

Percursos em Árvores Binárias

Estrutura de Dados — QXD0010



UNIVERSIDADE
FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

Prof. Atílio Gomes Luiz
gomes.atilio@ufc.br

Universidade Federal do Ceará

2º semestre/2022



Introdução



Percursos em Árvores Binárias

- Muitas operações em árvores binárias envolvem o percurso de todas as subárvores, com execução de alguma ação de tratamento em cada nó.

Percursos em Árvores Binárias

- Muitas operações em árvores binárias envolvem o percurso de todas as subárvores, com execução de alguma ação de tratamento em cada nó.
- É comum percorrer uma árvore em uma das seguintes ordens:
 - **pré-ordem**:
 - trata raiz, percorre `r->left`, percorre `r->right`

Percursos em Árvores Binárias

- Muitas operações em árvores binárias envolvem o percurso de todas as subárvores, com execução de alguma ação de tratamento em cada nó.
- É comum percorrer uma árvore em uma das seguintes ordens:
 - **pré-ordem:**
 - trata raiz, percorre `r->left`, percorre `r->right`
 - **ordem simétrica:**
 - percorre `r->left`, trata raiz, percorre `r->right`

Percursos em Árvores Binárias

- Muitas operações em árvores binárias envolvem o percurso de todas as subárvores, com execução de alguma ação de tratamento em cada nó.
- É comum percorrer uma árvore em uma das seguintes ordens:
 - **pré-ordem:**
 - trata raiz, percorre `r->left`, percorre `r->right`
 - **ordem simétrica:**
 - percorre `r->left`, trata raiz, percorre `r->right`
 - **pós-ordem:**
 - percorre `r->left`, percorre `r->right`, trata raiz

Percorrendo os nós — Pré-ordem

A pré-ordem

Percorrendo os nós — Pré-ordem

A pré-ordem

- primeiro visita (processa) a raiz

Percorrendo os nós — Pré-ordem

A pré-ordem

- primeiro visita (processa) a raiz
- depois a subárvore esquerda

Percorrendo os nós — Pré-ordem

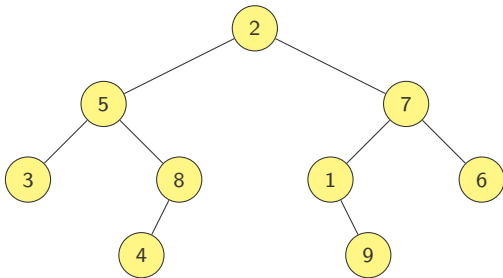
A pré-ordem

- primeiro visita (processa) a raiz
- depois a subárvore esquerda
- depois a subárvore direita

Percorrendo os nós — Pré-ordem

A pré-ordem

- primeiro visita (processa) a raiz
- depois a subárvore esquerda
- depois a subárvore direita

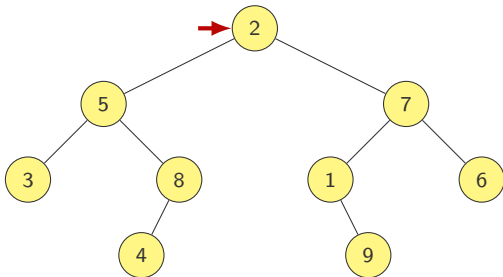


Ex:

Percorrendo os nós — Pré-ordem

A pré-ordem

- primeiro visita (processa) a raiz
- depois a subárvore esquerda
- depois a subárvore direita

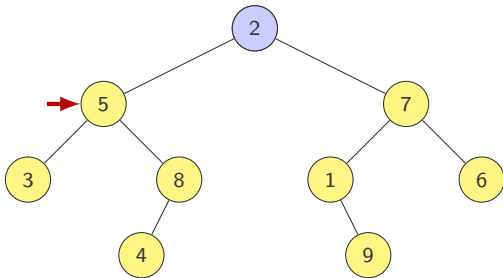


Ex:

Percorrendo os nós — Pré-ordem

A pré-ordem

- primeiro visita (processa) a raiz
- depois a subárvore esquerda
- depois a subárvore direita

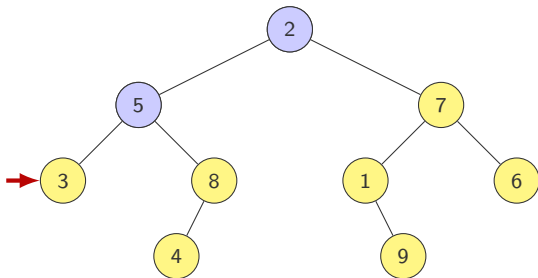


Ex: 2,

Percorrendo os nós — Pré-ordem

A pré-ordem

- primeiro visita (processa) a raiz
- depois a subárvore esquerda
- depois a subárvore direita

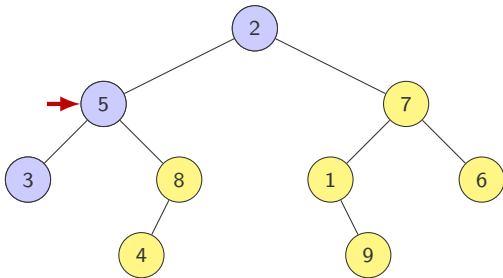


Ex: 2, 5,

Percorrendo os nós — Pré-ordem

A pré-ordem

- primeiro visita (processa) a raiz
- depois a subárvore esquerda
- depois a subárvore direita

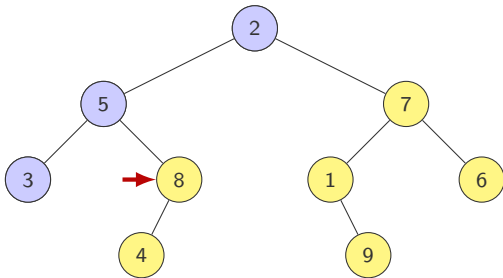


Ex: 2, 5, 3,

Percorrendo os nós — Pré-ordem

A pré-ordem

- primeiro visita (processa) a raiz
- depois a subárvore esquerda
- depois a subárvore direita

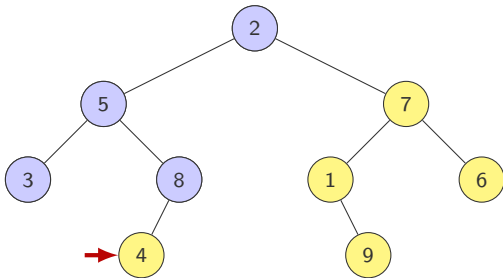


Ex: 2, 5, 3,

Percorrendo os nós — Pré-ordem

A pré-ordem

- primeiro visita (processa) a raiz
- depois a subárvore esquerda
- depois a subárvore direita

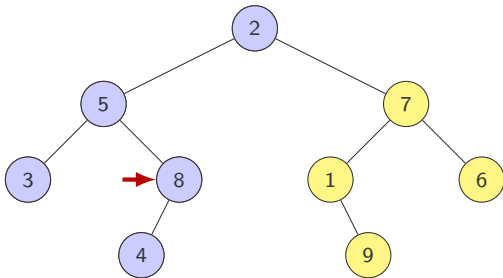


Ex: 2, 5, 3, 8,

Percorrendo os nós — Pré-ordem

A pré-ordem

- primeiro visita (processa) a raiz
- depois a subárvore esquerda
- depois a subárvore direita

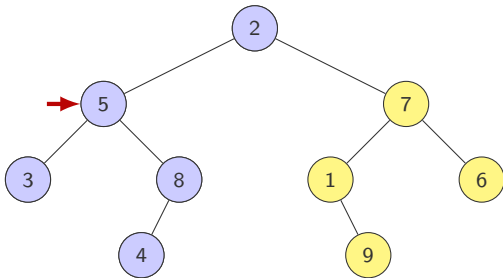


Ex: 2, 5, 3, 8, 4,

Percorrendo os nós — Pré-ordem

A pré-ordem

- primeiro visita (processa) a raiz
- depois a subárvore esquerda
- depois a subárvore direita

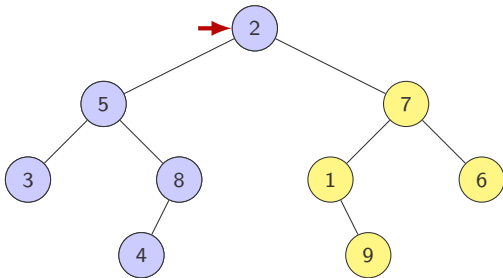


Ex: 2, 5, 3, 8, 4,

Percorrendo os nós — Pré-ordem

A pré-ordem

- primeiro visita (processa) a raiz
- depois a subárvore esquerda
- depois a subárvore direita

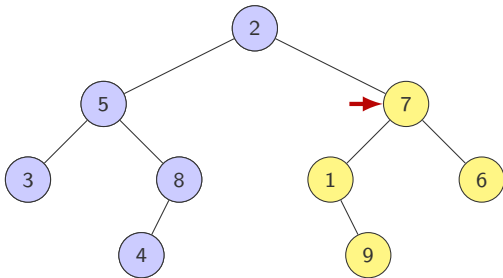


Ex: 2, 5, 3, 8, 4,

Percorrendo os nós — Pré-ordem

A pré-ordem

- primeiro visita (processa) a raiz
- depois a subárvore esquerda
- depois a subárvore direita

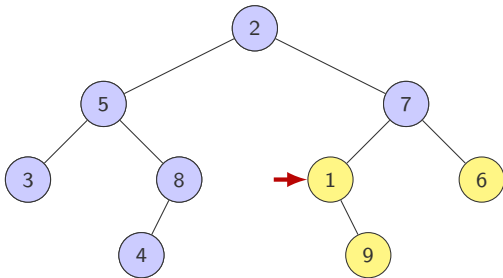


Ex: 2, 5, 3, 8, 4,

Percorrendo os nós — Pré-ordem

A pré-ordem

- primeiro visita (processa) a raiz
- depois a subárvore esquerda
- depois a subárvore direita

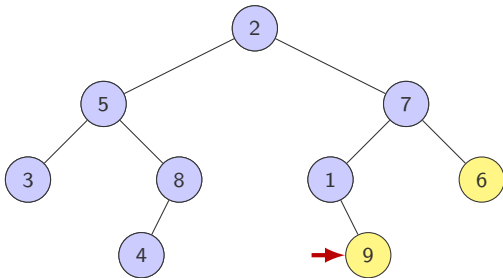


Ex: 2, 5, 3, 8, 4, 7,

Percorrendo os nós — Pré-ordem

A pré-ordem

- primeiro visita (processa) a raiz
- depois a subárvore esquerda
- depois a subárvore direita

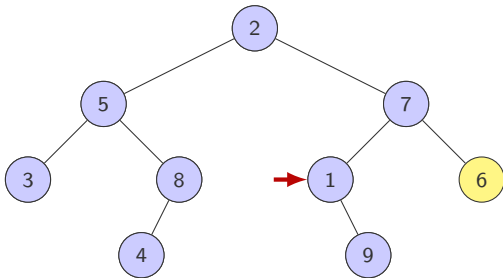


Ex: 2, 5, 3, 8, 4, 7, 1,

Percorrendo os nós — Pré-ordem

A pré-ordem

- primeiro visita (processa) a raiz
- depois a subárvore esquerda
- depois a subárvore direita

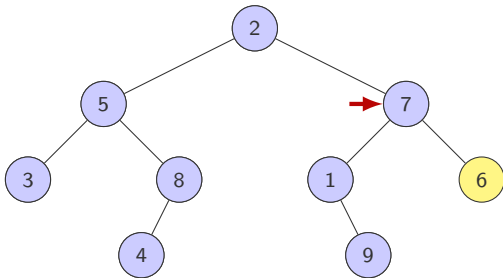


Ex: 2, 5, 3, 8, 4, 7, 1, 9,

Percorrendo os nós — Pré-ordem

A pré-ordem

- primeiro visita (processa) a raiz
- depois a subárvore esquerda
- depois a subárvore direita

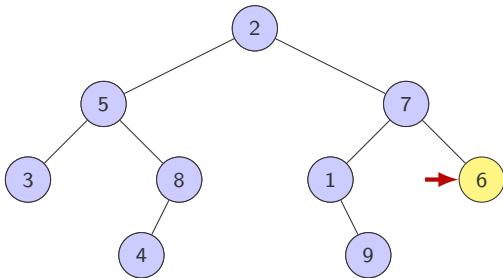


Ex: 2, 5, 3, 8, 4, 7, 1, 9,

Percorrendo os nós — Pré-ordem

A pré-ordem

- primeiro visita (processa) a raiz
- depois a subárvore esquerda
- depois a subárvore direita

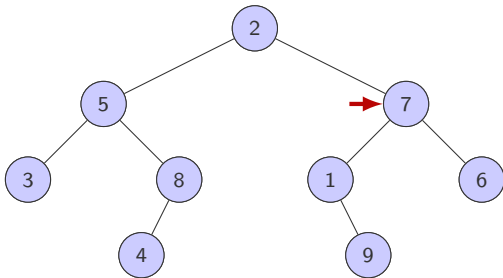


Ex: 2, 5, 3, 8, 4, 7, 1, 9,

Percorrendo os nós — Pré-ordem

A pré-ordem

- primeiro visita (processa) a raiz
- depois a subárvore esquerda
- depois a subárvore direita

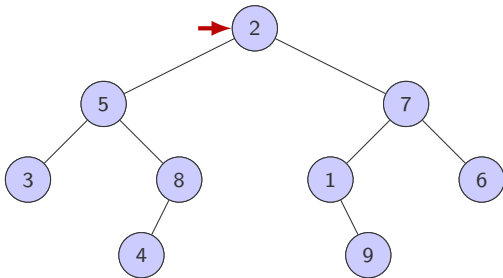


Ex: 2, 5, 3, 8, 4, 7, 1, 9, 6

Percorrendo os nós — Pré-ordem

A pré-ordem

- primeiro visita (processa) a raiz
- depois a subárvore esquerda
- depois a subárvore direita



Ex: 2, 5, 3, 8, 4, 7, 1, 9, 6

Percorrendo os nós — Pós-ordem

A pós-ordem

Percorrendo os nós — Pós-ordem

A pós-ordem

- primeiro visita a subárvore esquerda

Percorrendo os nós — Pós-ordem

A pós-ordem

- primeiro visita a subárvore esquerda
- depois a subárvore direita

Percorrendo os nós — Pós-ordem

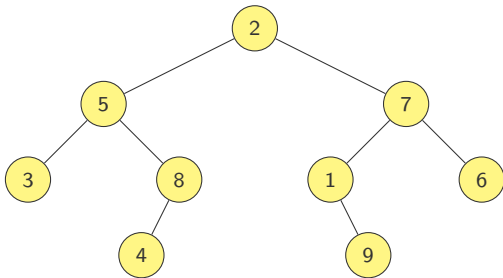
A pós-ordem

- primeiro visita a subárvore esquerda
- depois a subárvore direita
- e por último visita a raiz

Percorrendo os nós — Pós-ordem

A pós-ordem

- primeiro visita a subárvore esquerda
- depois a subárvore direita
- e por último visita a raiz

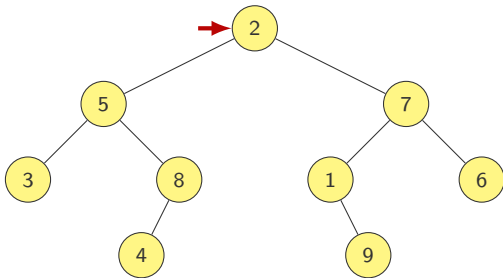


Ex:

Percorrendo os nós — Pós-ordem

A pós-ordem

- primeiro visita a subárvore esquerda
- depois a subárvore direita
- e por último visita a raiz

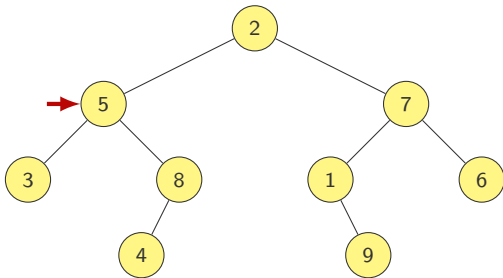


Ex:

Percorrendo os nós — Pós-ordem

A pós-ordem

- primeiro visita a subárvore esquerda
- depois a subárvore direita
- e por último visita a raiz

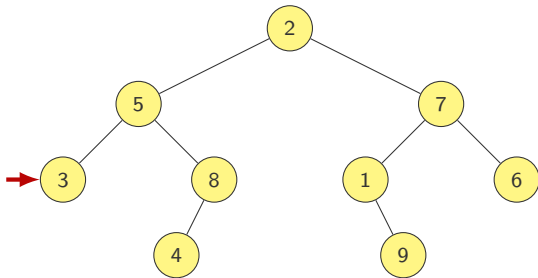


Ex:

Percorrendo os nós — Pós-ordem

A pós-ordem

- primeiro visita a subárvore esquerda
- depois a subárvore direita
- e por último visita a raiz

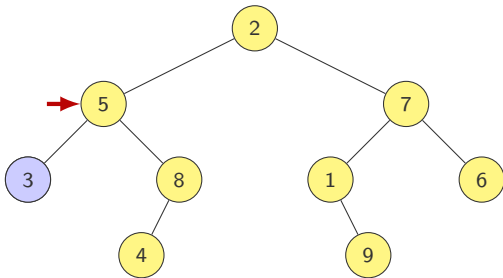


Ex:

Percorrendo os nós — Pós-ordem

A pós-ordem

- primeiro visita a subárvore esquerda
- depois a subárvore direita
- e por último visita a raiz

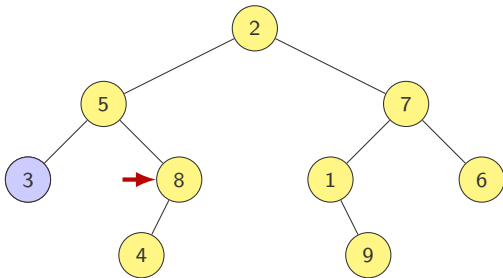


Ex: 3,

Percorrendo os nós — Pós-ordem

A pós-ordem

- primeiro visita a subárvore esquerda
- depois a subárvore direita
- e por último visita a raiz

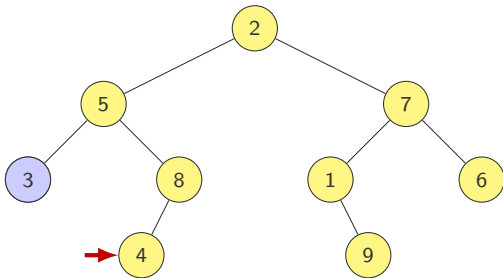


Ex: 3,

Percorrendo os nós — Pós-ordem

A pós-ordem

- primeiro visita a subárvore esquerda
- depois a subárvore direita
- e por último visita a raiz

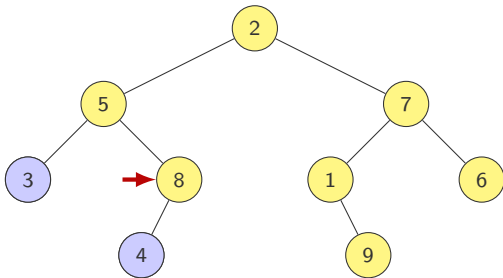


Ex: 3,

Percorrendo os nós — Pós-ordem

A pós-ordem

- primeiro visita a subárvore esquerda
- depois a subárvore direita
- e por último visita a raiz

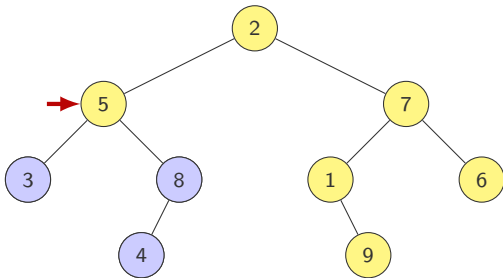


Ex: 3, 4,

Percorrendo os nós — Pós-ordem

A pós-ordem

- primeiro visita a subárvore esquerda
- depois a subárvore direita
- e por último visita a raiz

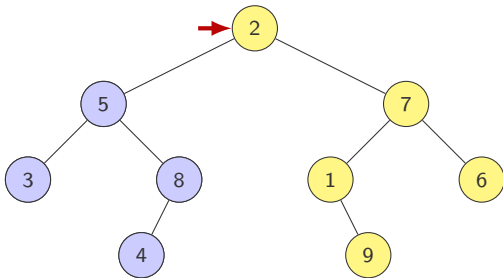


Ex: 3, 4, 8,

Percorrendo os nós — Pós-ordem

A pós-ordem

- primeiro visita a subárvore esquerda
- depois a subárvore direita
- e por último visita a raiz

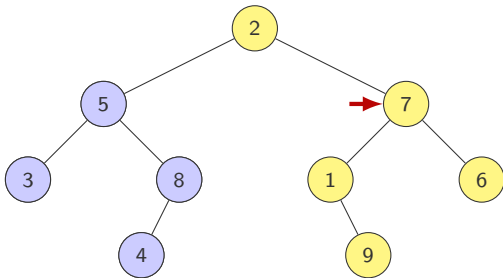


Ex: 3, 4, 8, 5,

Percorrendo os nós — Pós-ordem

A pós-ordem

- primeiro visita a subárvore esquerda
- depois a subárvore direita
- e por último visita a raiz

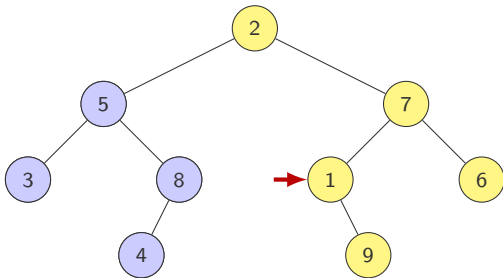


Ex: 3, 4, 8, 5,

Percorrendo os nós — Pós-ordem

A pós-ordem

- primeiro visita a subárvore esquerda
- depois a subárvore direita
- e por último visita a raiz

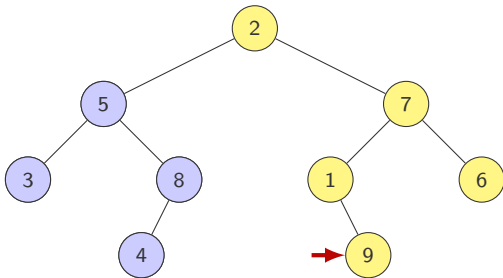


Ex: 3, 4, 8, 5,

Percorrendo os nós — Pós-ordem

A pós-ordem

- primeiro visita a subárvore esquerda
- depois a subárvore direita
- e por último visita a raiz

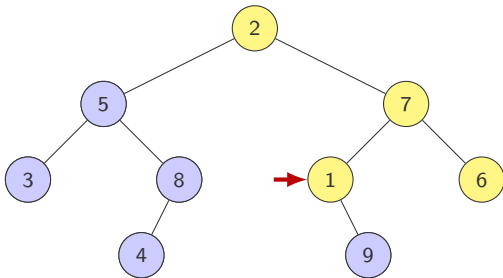


Ex: 3, 4, 8, 5,

Percorrendo os nós — Pós-ordem

A pós-ordem

- primeiro visita a subárvore esquerda
- depois a subárvore direita
- e por último visita a raiz

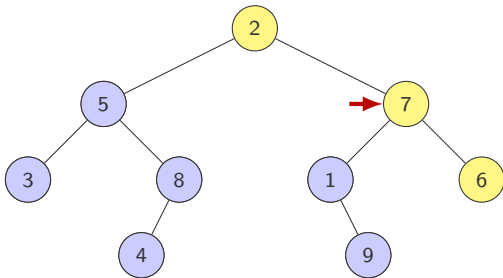


Ex: 3, 4, 8, 5, 9,

Percorrendo os nós — Pós-ordem

A pós-ordem

- primeiro visita a subárvore esquerda
- depois a subárvore direita
- e por último visita a raiz

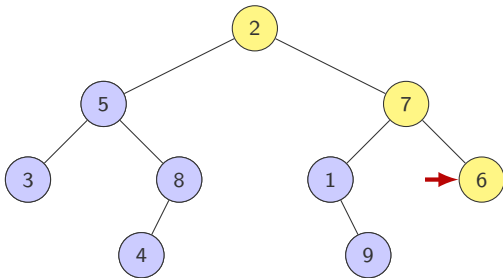


Ex: 3, 4, 8, 5, 9, 1,

Percorrendo os nós — Pós-ordem

A pós-ordem

- primeiro visita a subárvore esquerda
- depois a subárvore direita
- e por último visita a raiz

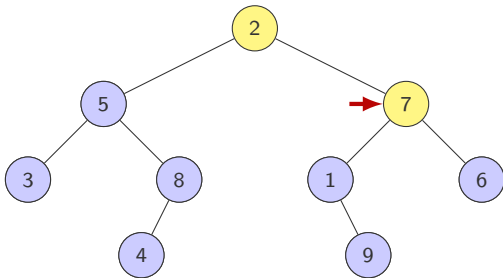


Ex: 3, 4, 8, 5, 9, 1,

Percorrendo os nós — Pós-ordem

A pós-ordem

- primeiro visita a subárvore esquerda
- depois a subárvore direita
- e por último visita a raiz

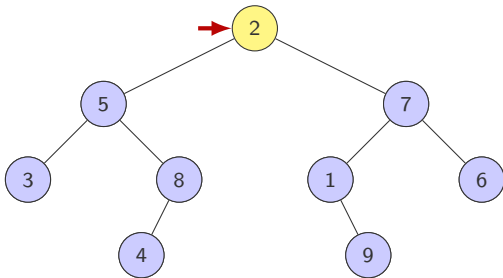


Ex: 3, 4, 8, 5, 9, 1, 6,

Percorrendo os nós — Pós-ordem

A pós-ordem

- primeiro visita a subárvore esquerda
- depois a subárvore direita
- e por último visita a raiz

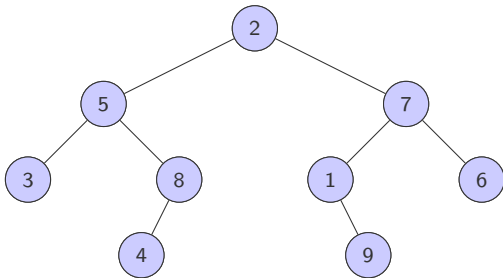


Ex: 3, 4, 8, 5, 9, 1, 6, 7,

Percorrendo os nós — Pós-ordem

A pós-ordem

- primeiro visita a subárvore esquerda
- depois a subárvore direita
- e por último visita a raiz



Ex: 3, 4, 8, 5, 9, 1, 6, 7, 2

Percorrendo os nós — Ordem Simétrica

A ordem simétrica

Percorrendo os nós — Ordem Simétrica

A ordem simétrica

- primeiro visita a subárvore esquerda

Percorrendo os nós — Ordem Simétrica

A ordem simétrica

- primeiro visita a subárvore esquerda
- depois visita a raiz

Percorrendo os nós — Ordem Simétrica

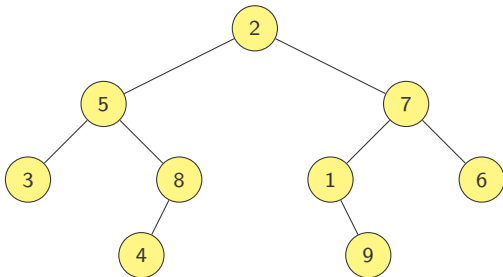
A ordem simétrica

- primeiro visita a subárvore esquerda
- depois visita a raiz
- e por última visita a subárvore direita

Percorrendo os nós — Ordem Simétrica

A ordem simétrica

- primeiro visita a subárvore esquerda
- depois visita a raiz
- e por última visita a subárvore direita

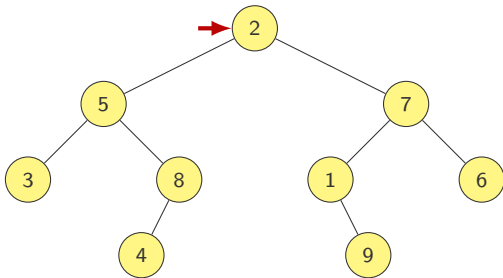


Ex:

Percorrendo os nós — Ordem Simétrica

A ordem simétrica

- primeiro visita a subárvore esquerda
- depois visita a raiz
- e por última visita a subárvore direita

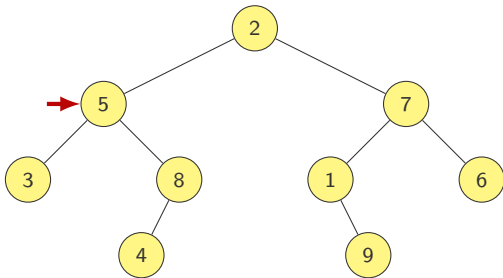


Ex:

Percorrendo os nós — Ordem Simétrica

A ordem simétrica

- primeiro visita a subárvore esquerda
- depois visita a raiz
- e por última visita a subárvore direita

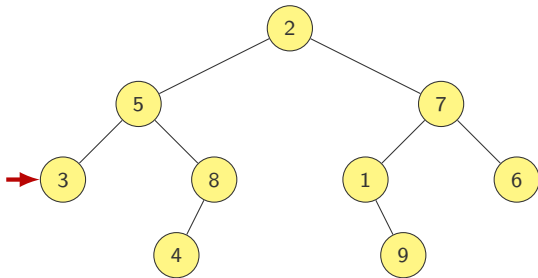


Ex:

Percorrendo os nós — Ordem Simétrica

A ordem simétrica

- primeiro visita a subárvore esquerda
- depois visita a raiz
- e por última visita a subárvore direita

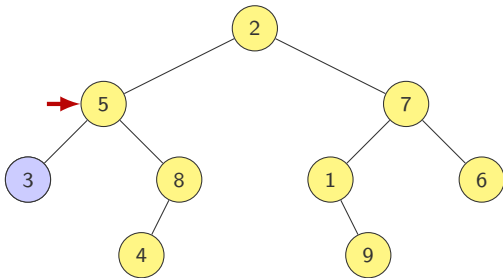


Ex:

Percorrendo os nós — Ordem Simétrica

A ordem simétrica

- primeiro visita a subárvore esquerda
- depois visita a raiz
- e por última visita a subárvore direita

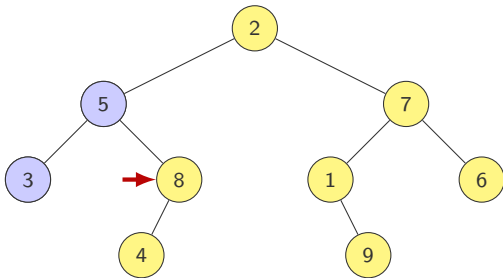


Ex: 3,

Percorrendo os nós — Ordem Simétrica

A ordem simétrica

- primeiro visita a subárvore esquerda
- depois visita a raiz
- e por última visita a subárvore direita

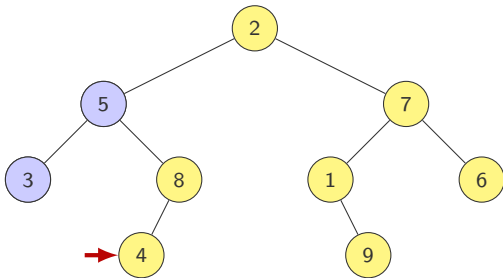


Ex: 3, 5,

Percorrendo os nós — Ordem Simétrica

A ordem simétrica

- primeiro visita a subárvore esquerda
- depois visita a raiz
- e por última visita a subárvore direita

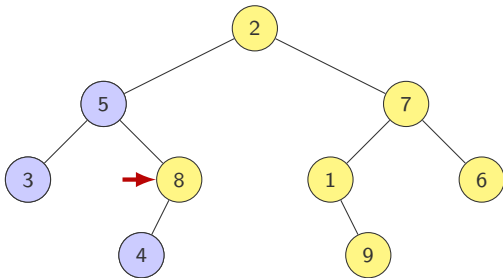


Ex: 3, 5,

Percorrendo os nós — Ordem Simétrica

A ordem simétrica

- primeiro visita a subárvore esquerda
- depois visita a raiz
- e por última visita a subárvore direita

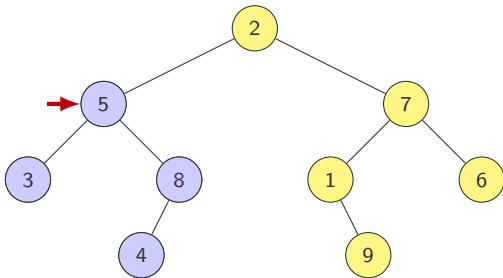


Ex: 3, 5, 4,

Percorrendo os nós — Ordem Simétrica

A ordem simétrica

- primeiro visita a subárvore esquerda
- depois visita a raiz
- e por última visita a subárvore direita

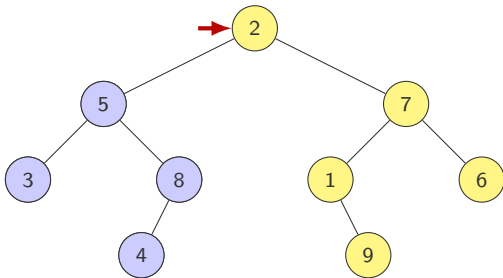


Ex: 3, 5, 4, 8,

Percorrendo os nós — Ordem Simétrica

A ordem simétrica

- primeiro visita a subárvore esquerda
- depois visita a raiz
- e por última visita a subárvore direita

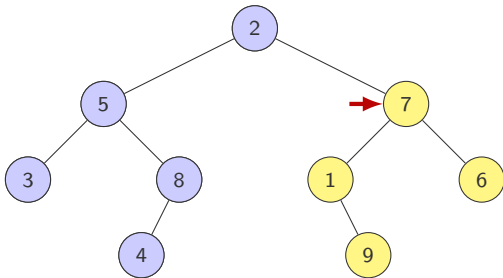


Ex: 3, 5, 4, 8,

Percorrendo os nós — Ordem Simétrica

A ordem simétrica

- primeiro visita a subárvore esquerda
- depois visita a raiz
- e por última visita a subárvore direita

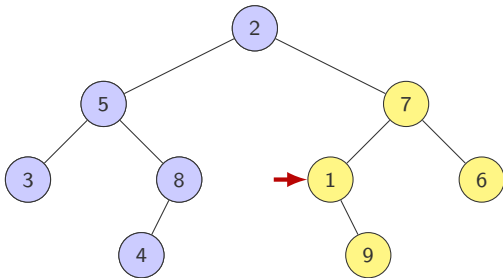


Ex: 3, 5, 4, 8, 2,

Percorrendo os nós — Ordem Simétrica

A ordem simétrica

- primeiro visita a subárvore esquerda
- depois visita a raiz
- e por última visita a subárvore direita

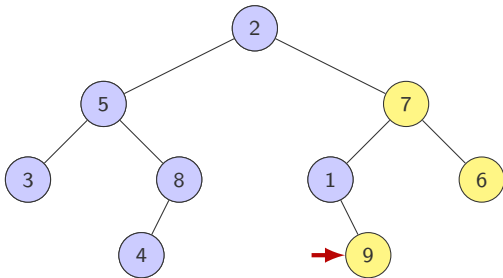


Ex: 3, 5, 4, 8, 2,

Percorrendo os nós — Ordem Simétrica

A ordem simétrica

- primeiro visita a subárvore esquerda
- depois visita a raiz
- e por última visita a subárvore direita

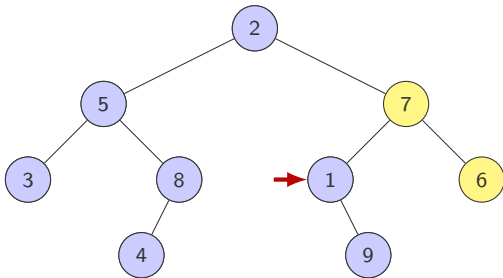


Ex: 3, 5, 4, 8, 2, 1,

Percorrendo os nós — Ordem Simétrica

A ordem simétrica

- primeiro visita a subárvore esquerda
- depois visita a raiz
- e por última visita a subárvore direita

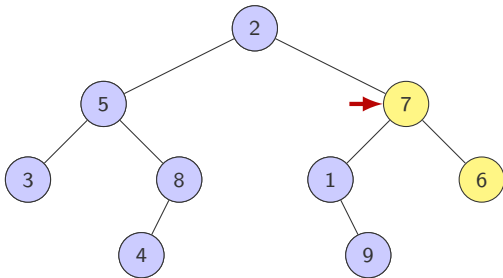


Ex: 3, 5, 4, 8, 2, 1, 9,

Percorrendo os nós — Ordem Simétrica

A ordem simétrica

- primeiro visita a subárvore esquerda
- depois visita a raiz
- e por última visita a subárvore direita

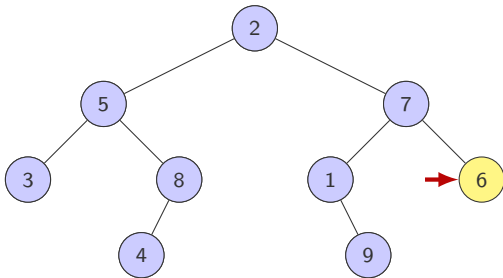


Ex: 3, 5, 4, 8, 2, 1, 9,

Percorrendo os nós — Ordem Simétrica

A ordem simétrica

- primeiro visita a subárvore esquerda
- depois visita a raiz
- e por última visita a subárvore direita

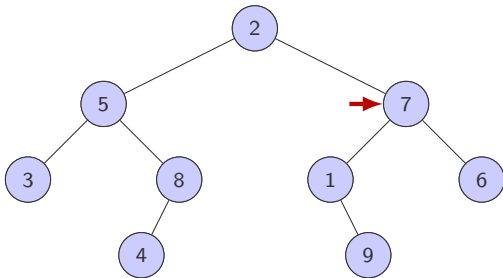


Ex: 3, 5, 4, 8, 2, 1, 9, 7,

Percorrendo os nós — Ordem Simétrica

A ordem simétrica

- primeiro visita a subárvore esquerda
- depois visita a raiz
- e por última visita a subárvore direita

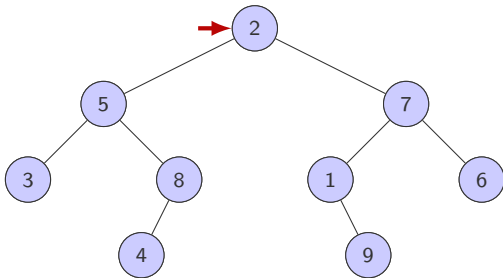


Ex: 3, 5, 4, 8, 2, 1, 9, 7, 6

Percorrendo os nós — Ordem Simétrica

A ordem simétrica

- primeiro visita a subárvore esquerda
- depois visita a raiz
- e por última visita a subárvore direita



Ex: 3, 5, 4, 8, 2, 1, 9, 7, 6

Implementação de percurso em profundidade

```
1 void Tree::preOrdem(Node *node) {
2     if(node != nullptr) {
3         std::cout << node->key << " "; // visita a raiz
4         preOrdem(node->left);
5         preOrdem(node->right);
6     }
7 }
8
9 void Tree::ordemSimetrica(Node *node) {
10     if(node != nullptr) {
11         ordemSimetrica(node->left);
12         std::cout << node->key << " "; // visita a raiz
13         ordemSimetrica(node->right);
14     }
15 }
16
17 void Tree::posOrdem(Node *node) {
18     if(node != nullptr) {
19         posOrdem(node->left);
20         posOrdem(node->right);
21         std::cout << node->key << " "; // visita a raiz
22     }
23 }
```

Percursos Iterativos em Árvore



Percurso em pré-ordem — Recursivo

Vimos em slides anteriores que o percurso em pré-ordem recursivo é implementado pela seguinte função:

```
1 void Tree::preOrdem(Node *node) {  
2     if(node != nullptr) {  
3         std::cout << node->key << " "; // visita a raiz  
4         preOrdem(node->left);  
5         preOrdem(node->right);  
6     }  
7 }
```

Como implementar a pré-ordem sem usar recursão?

Percurso em pré-ordem — Iterativo

Vamos usar a class `stack` da biblioteca STL do C++

```
1 void Tree::preOrdemIterativa() {
2     std::stack<Node*> pilha; // cria pilha vazia
3     pilha.push(root); // empilha raiz
4
5     while(!pilha.empty()) {
6         Node *node = pilha.top();
7         pilha.pop();
8         if(node != nullptr) {
9             std::cout << node->key << " "; // visita raiz
10            pilha.push(node->right);
11            pilha.push(node->left);
12        }
13    }
14 }
```

Percurso em pré-ordem — Iterativo

Vamos usar a class `stack` da biblioteca STL do C++

```
1 void Tree::preOrdemIterativa() {
2     std::stack<Node*> pilha; // cria pilha vazia
3     pilha.push(root); // empilha raiz
4
5     while(!pilha.empty()) {
6         Node *node = pilha.top();
7         pilha.pop();
8         if(node != nullptr) {
9             std::cout << node->key << " "; // visita raiz
10            pilha.push(node->right);
11            pilha.push(node->left);
12        }
13    }
14 }
```

Por que empilhamos `node->right` primeiro?

Percurso em pré-ordem — Iterativo

Vamos usar a class `stack` da biblioteca STL do C++

```
1 void Tree::preOrdemIterativa() {
2     std::stack<Node*> pilha; // cria pilha vazia
3     pilha.push(root); // empilha raiz
4
5     while(!pilha.empty()) {
6         Node *node = pilha.top();
7         pilha.pop();
8         if(node != nullptr) {
9             std::cout << node->key << " "; // visita raiz
10            pilha.push(node->right);
11            pilha.push(node->left);
12        }
13    }
14 }
```

Por que empilhamos `node->right` primeiro?

- E se fosse o contrário?

Percurso em ordem simétrica — Recursivo

O percurso em ordem simétrica (inordem) recursivo é implementado pela seguinte função:

```
1 void Tree::ordemSimetrica(Node *node) {  
2     if(node != nullptr) {  
3         ordemSimetrica(node->left);  
4         std::cout << node->key << " "; // visita a raiz  
5         ordemSimetrica(node->right);  
6     }  
7 }
```

Como percorrer em ordem simétrica sem usar recursão?

Percurso em ordem simétrica — Iterativo

Vamos usar a class `stack` da biblioteca STL do C++

```
1 void Tree::ordemSimetricaIterativa() {
2     std::stack<Node*> pilha;
3     Node *node = root;
4
5     while(!pilha.empty() || node != nullptr) {
6         if(node != nullptr) {
7             pilha.push(node);
8             node = node->left;
9         }
10        else {
11            node = pilha.top();
12            pilha.pop();
13            std::cout << node->key << " ";
14            node = node->right;
15        }
16    }
17 }
```

Percurso em ordem simétrica — Iterativo

- **Exercício para casa:** Implementar o percurso em pós-ordem iterativo. (Este percurso é um pouco mais complexo do que os anteriores)

Percurso em largura



Percorrendo os nós (em largura)

O percurso em largura

Percorrendo os nós (em largura)

O percurso em largura

- visita os nós por níveis

Percorrendo os nós (em largura)

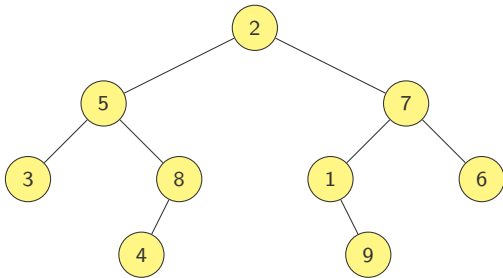
O percurso em largura

- visita os nós por níveis
- da esquerda para a direita

Percorrendo os nós (em largura)

O percurso em largura

- visita os nós por níveis
- da esquerda para a direita

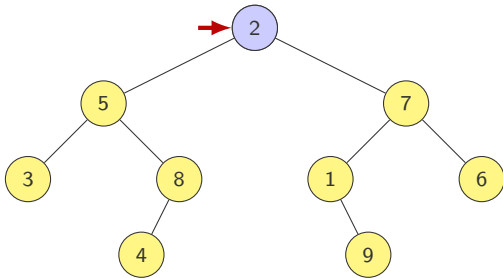


Ex:

Percorrendo os nós (em largura)

O percurso em largura

- visita os nós por níveis
- da esquerda para a direita

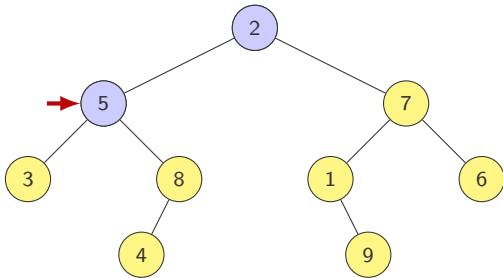


Ex: 2,

Percorrendo os nós (em largura)

O percurso em largura

- visita os nós por níveis
- da esquerda para a direita

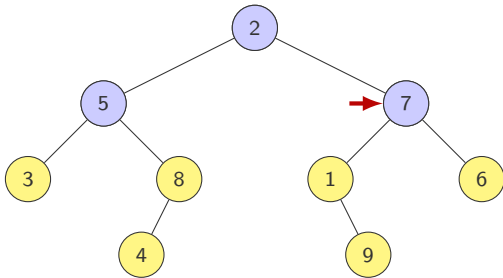


Ex: 2, 5,

Percorrendo os nós (em largura)

O percurso em largura

- visita os nós por níveis
- da esquerda para a direita

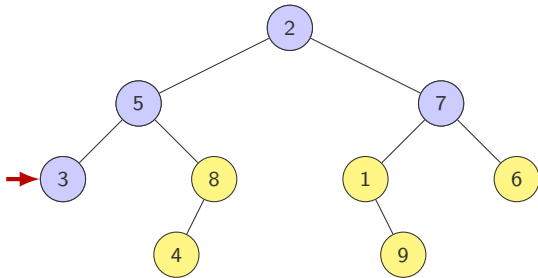


Ex: 2, 5, 7,

Percorrendo os nós (em largura)

O percurso em largura

- visita os nós por níveis
- da esquerda para a direita

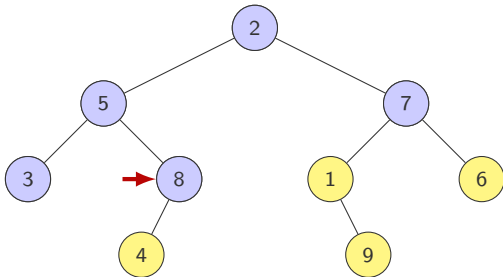


Ex: 2, 5, 7, 3,

Percorrendo os nós (em largura)

O percurso em largura

- visita os nós por níveis
- da esquerda para a direita

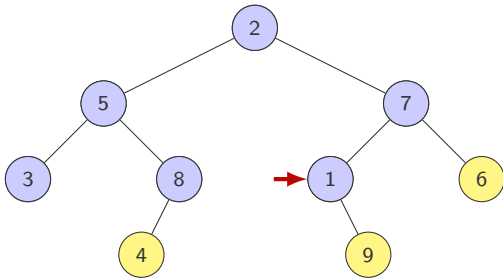


Ex: 2, 5, 7, 3, 8,

Percorrendo os nós (em largura)

O percurso em largura

- visita os nós por níveis
- da esquerda para a direita

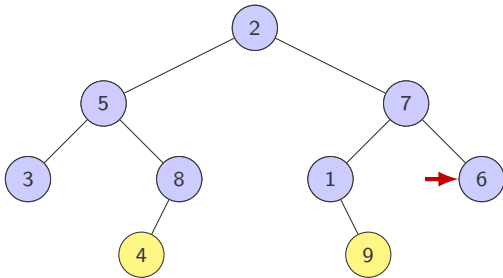


Ex: 2, 5, 7, 3, 8, 1,

Percorrendo os nós (em largura)

O percurso em largura

- visita os nós por níveis
- da esquerda para a direita

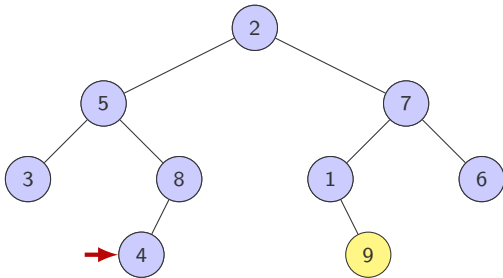


Ex: 2, 5, 7, 3, 8, 1, 6,

Percorrendo os nós (em largura)

O percurso em largura

- visita os nós por níveis
- da esquerda para a direita

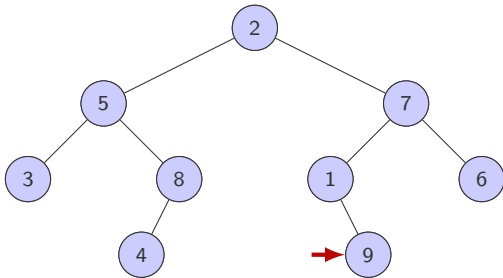


Ex: 2, 5, 7, 3, 8, 1, 6, 4,

Percorrendo os nós (em largura)

O percurso em largura

- visita os nós por níveis
- da esquerda para a direita



Ex: 2, 5, 7, 3, 8, 1, 6, 4, 9

Implementação do percurso em largura

Como implementar a busca em largura?

Implementação do percurso em largura

Como implementar a busca em largura?

- Usamos uma fila

Implementação do percurso em largura

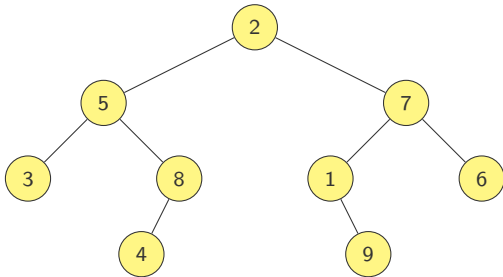
Como implementar a busca em largura?

- Usamos uma fila
- Colocamos a raiz na fila e depois

Implementação do percurso em largura

Como implementar a busca em largura?

- Usamos uma fila
- Colocamos a raiz na fila e depois
- pegamos um elemento da fila e enfileiramos seus filhos

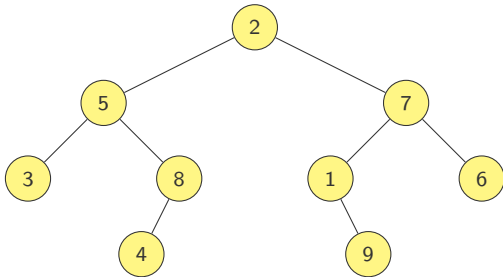


Fila

Implementação do percurso em largura

Como implementar a busca em largura?

- Usamos uma fila
- Colocamos a raiz na fila e depois
- pegamos um elemento da fila e enfileiramos seus filhos



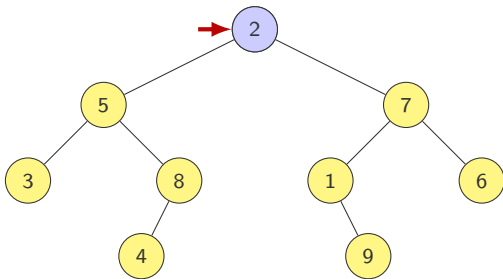
Fila

2

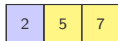
Implementação do percurso em largura

Como implementar a busca em largura?

- Usamos uma fila
- Colocamos a raiz na fila e depois
- pegamos um elemento da fila e enfileiramos seus filhos



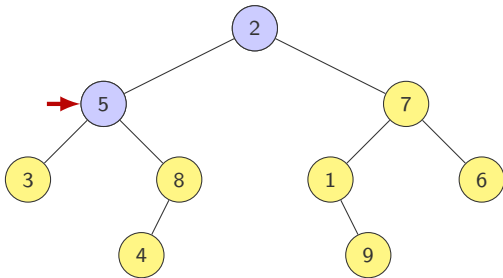
Fila



Implementação do percurso em largura

Como implementar a busca em largura?

- Usamos uma fila
- Colocamos a raiz na fila e depois
- pegamos um elemento da fila e enfileiramos seus filhos



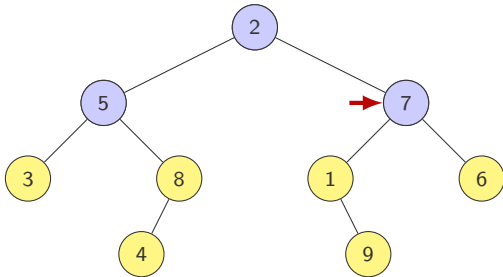
Fila

2	5	7	3	8
---	---	---	---	---

Implementação do percurso em largura

Como implementar a busca em largura?

- Usamos uma fila
- Colocamos a raiz na fila e depois
- pegamos um elemento da fila e enfileiramos seus filhos



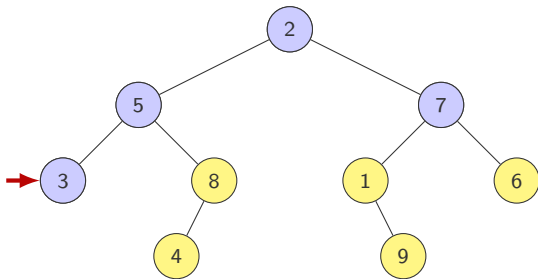
Fila

2	5	7	3	8	1	6
---	---	---	---	---	---	---

Implementação do percurso em largura

Como implementar a busca em largura?

- Usamos uma fila
- Colocamos a raiz na fila e depois
- pegamos um elemento da fila e enfileiramos seus filhos



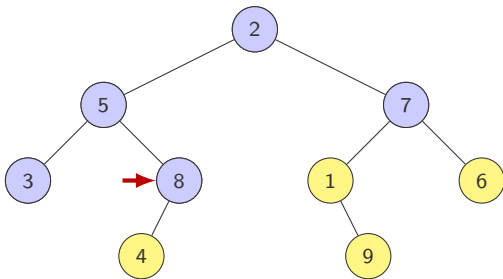
Fila

2	5	7	3	8	1	6
---	---	---	---	---	---	---

Implementação do percurso em largura

Como implementar a busca em largura?

- Usamos uma fila
- Colocamos a raiz na fila e depois
- pegamos um elemento da fila e enfileiramos seus filhos



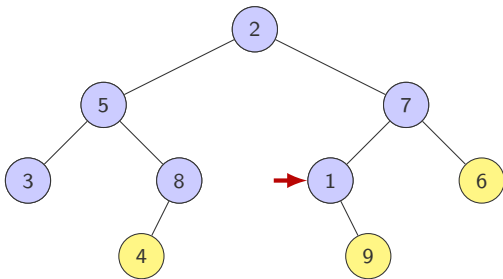
Fila



Implementação do percurso em largura

Como implementar a busca em largura?

- Usamos uma fila
- Colocamos a raiz na fila e depois
- pegamos um elemento da fila e enfileiramos seus filhos



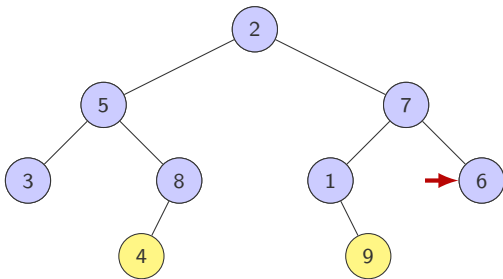
Fila

2	5	7	3	8	1	6	4	9
---	---	---	---	---	---	---	---	---

Implementação do percurso em largura

Como implementar a busca em largura?

- Usamos uma fila
- Colocamos a raiz na fila e depois
- pegamos um elemento da fila e enfileiramos seus filhos



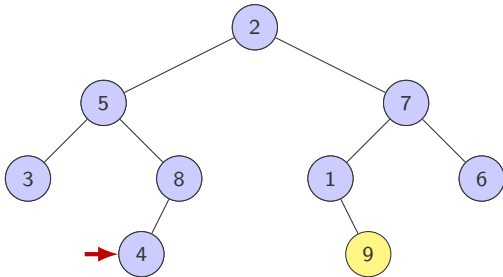
Fila

2	5	7	3	8	1	6	4	9
---	---	---	---	---	---	---	---	---

Implementação do percurso em largura

Como implementar a busca em largura?

- Usamos uma fila
- Colocamos a raiz na fila e depois
- pegamos um elemento da fila e enfileiramos seus filhos



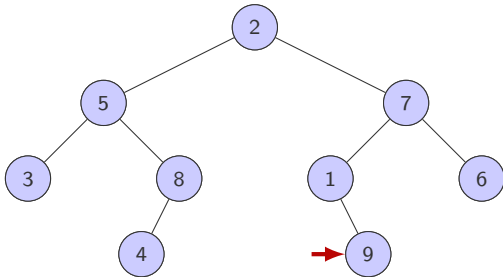
Fila

2	5	7	3	8	1	6	4	9
---	---	---	---	---	---	---	---	---

Implementação do percurso em largura

Como implementar a busca em largura?

- Usamos uma fila
- Colocamos a raiz na fila e depois
- pegamos um elemento da fila e enfileiramos seus filhos



Fila

2	5	7	3	8	1	6	4	9
---	---	---	---	---	---	---	---	---

Percurso em largura

Vamos usar a class `queue` da biblioteca STL do C++

```
1 void Tree::levelTraversal() {
2     std::queue<Node*> fila; // cria fila vazia
3     fila.push(root); // enfileira a raiz na fila
4
5     while (!fila.empty()) {
6         Node *node = fila.front();
7         fila.pop();
8         if (node != nullptr) {
9             fila.push(node->left);
10            fila.push(node->right);
11            std::cout << node->key << " "; // visita raiz
12        }
13    }
14 }
```

Percurso em largura

Vamos usar a class `queue` da biblioteca STL do C++

```
1 void Tree::levelTraversal() {
2     std::queue<Node*> fila; // cria fila vazia
3     fila.push(root); // enfileira a raiz na fila
4
5     while (!fila.empty()) {
6         Node *node = fila.front();
7         fila.pop();
8         if (node != nullptr) {
9             fila.push(node->left);
10            fila.push(node->right);
11            std::cout << node->key << " "; // visita raiz
12        }
13    }
14 }
```

Agora enfileiramos `node->left` primeiro

Percurso em largura

Vamos usar a class `queue` da biblioteca STL do C++

```
1 void Tree::levelTraversal() {
2     std::queue<Node*> fila; // cria fila vazia
3     fila.push(root); // enfileira a raiz na fila
4
5     while (!fila.empty()) {
6         Node *node = fila.front();
7         fila.pop();
8         if (node != nullptr) {
9             fila.push(node->left);
10            fila.push(node->right);
11            std::cout << node->key << " "; // visita raiz
12        }
13    }
14 }
```

Agora enfileiramos `node->left` primeiro

- E se fosse o contrário?

Exercícios



Exercício

Exercício: Escreva uma função iterativa que calcula o número de nós de uma árvore. A função deve obedecer o seguinte protótipo: `int`

`bt_size_iterative(Node* node);`

- **Dica:** você vai precisar de uma pilha.

Exercício

Exercício: Escreva uma função iterativa que calcula a altura de uma árvore. A função deve obedecer o seguinte protótipo:

```
int bt_height_iterative(Node* node);
```

- **Dica:** Você pode usar uma fila.

FIM

