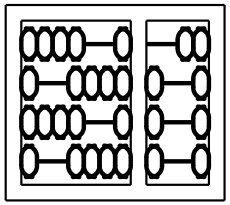


Universidade Estadual de Campinas - UNICAMP
Instituto de Computação - IC

Árvores: Parte 4

Tópicos

1. Rotações de subárvores
2. Árvores auto-ajustadas
3. Auto-reestruturação
4. Afunilamento
5. Semi-afunilamento

Rotações de subárvores

- Preservam o percurso em *inordem* da árvore original.

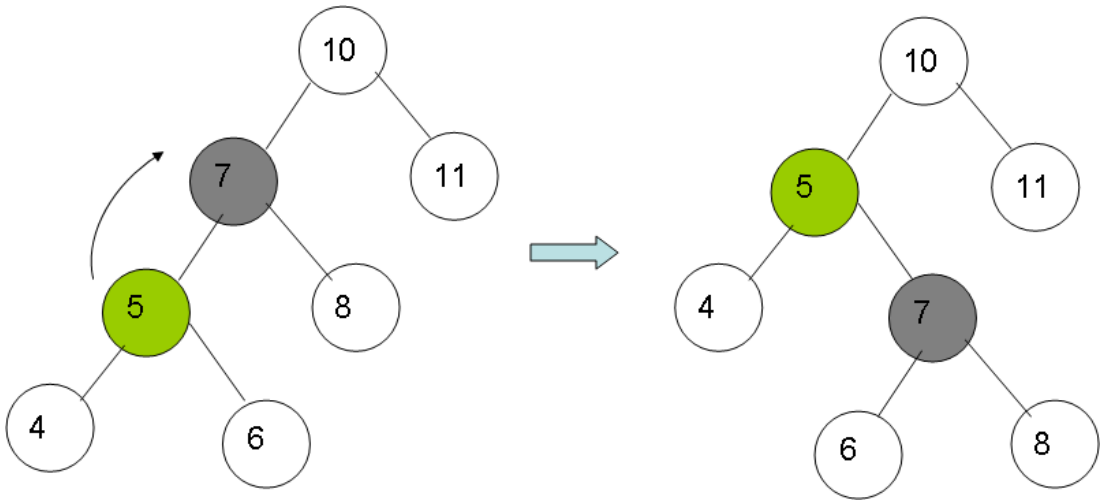
– Tipos de rotações:

1. À direita: rotaciona um dado nó, filho esquerdo, ao redor de seu ascendente.

2. À esquerda: rotaciona um dado nó, filho direito, ao redor de seu ascendente.

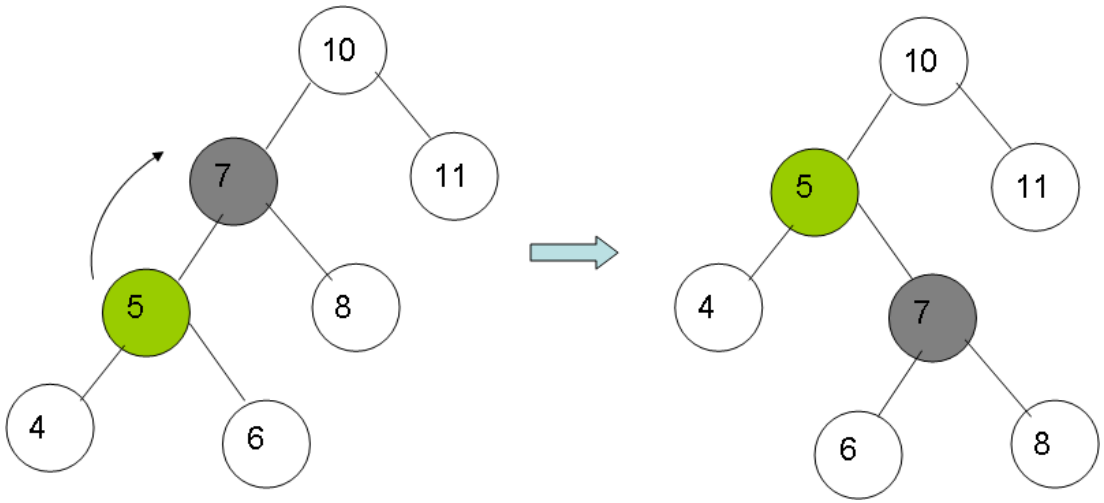
- Rotação à direita:

Exemplo 1:



- Rotação à direita:

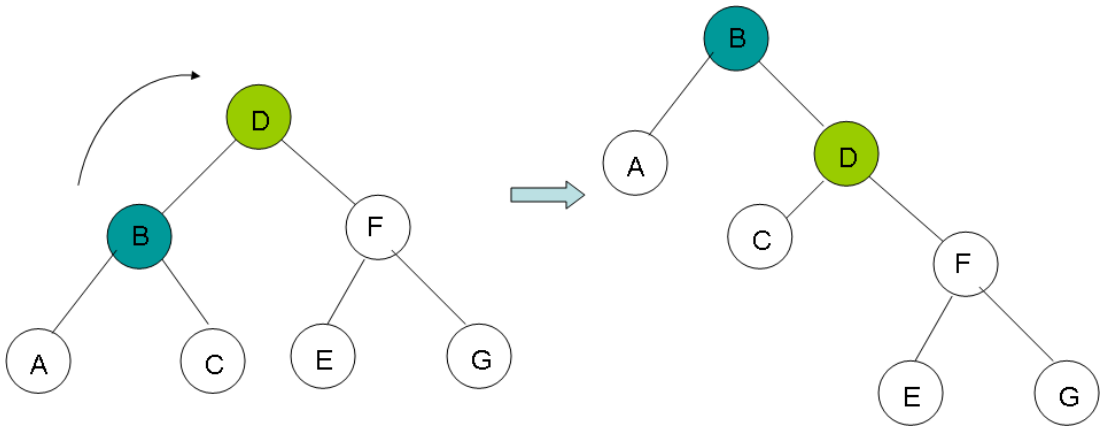
Exemplo 1:



– Percurso em *inordem*: *4 5 6 7 8 10 11*

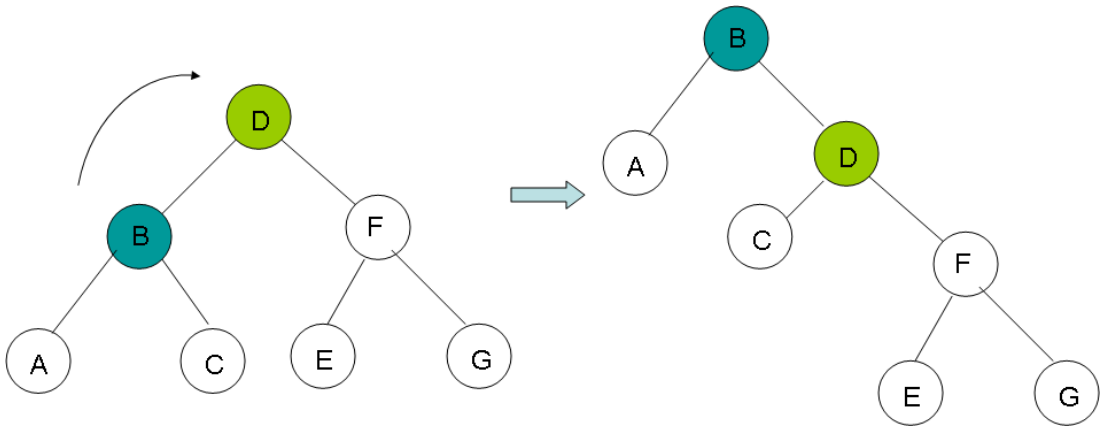
- Rotação à direita:

Exemplo 2:



- Rotação à direita:

Exemplo 2:

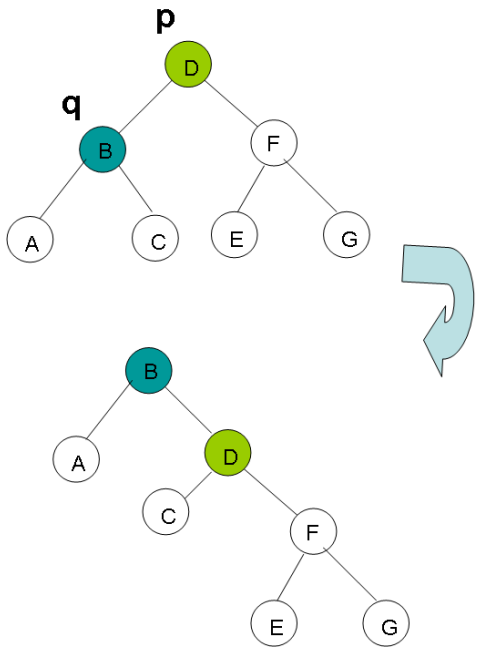


– Percurso em *inordem*: *A B C D E F G*

- Implementação:

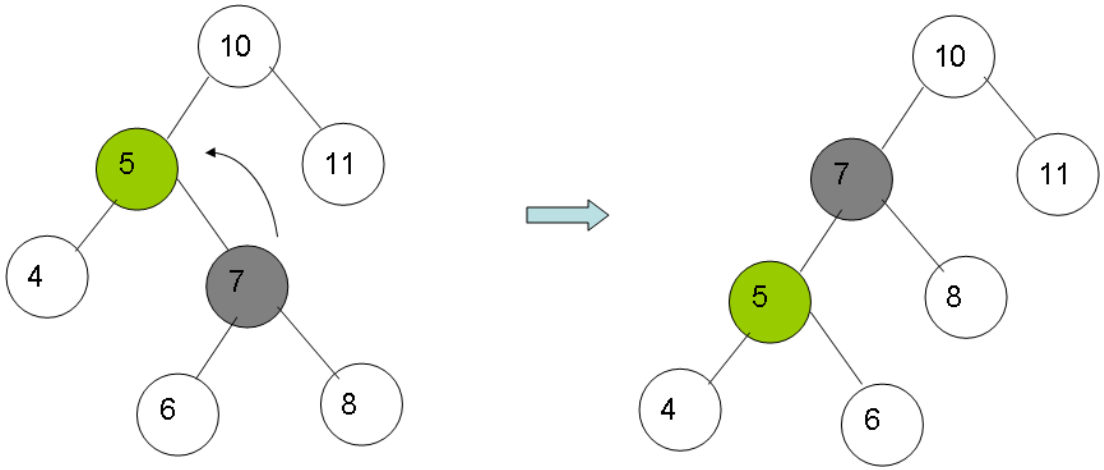
```
/* recebe no p e rotaciona filho esquerdo
    q. Retorna nó rotacionado */

void RotDir (NoArvBin *p, NoArvBin **q) {
  *q = p->esq ;
   p->esq = (*q)->dir ;
   (*q)->dir = p ;
}
```



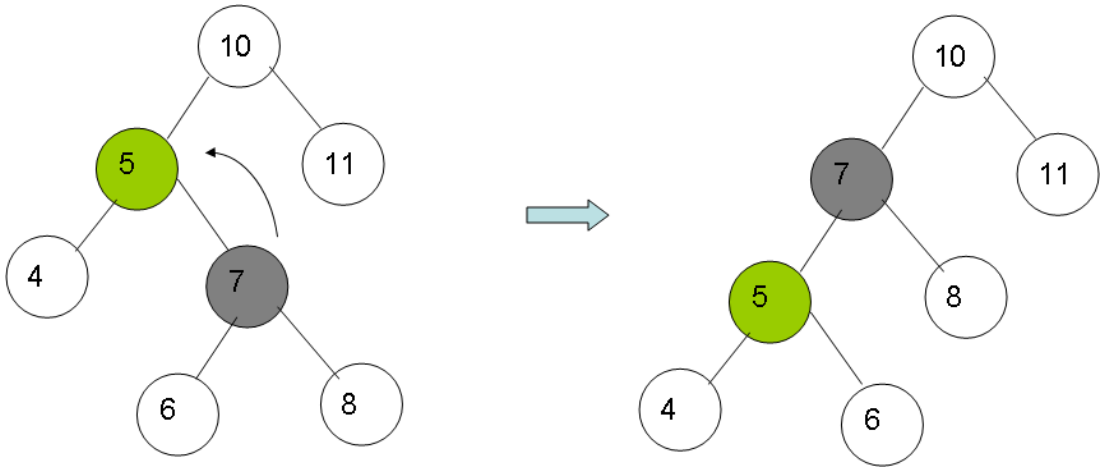
- Rotação à esquerda:

Exemplo 1:



- Rotação à esquerda:

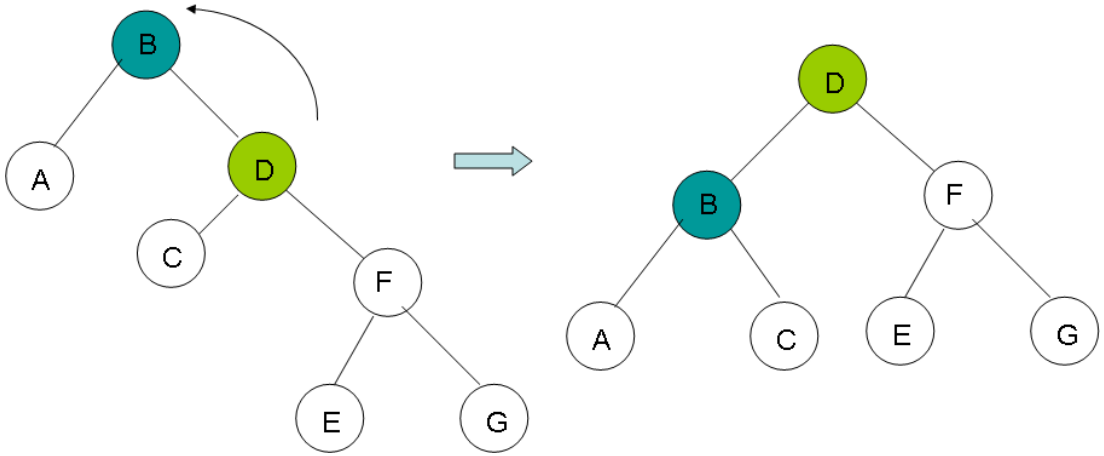
Exemplo 1:



– Percurso em *inordem*: *4 5 6 7 8 10 11*

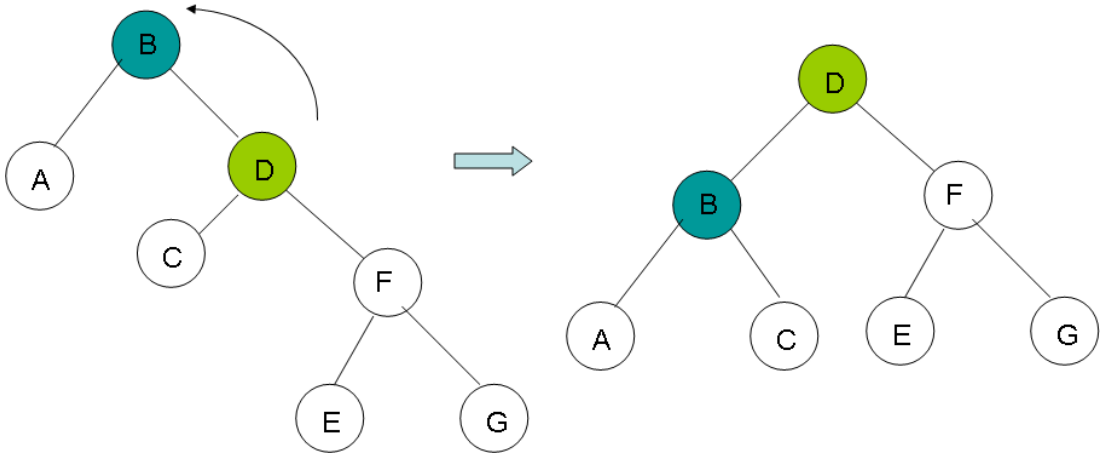
- Rotação à esquerda:

Exemplo 2:



- Rotação à esquerda:

Exemplo 2:

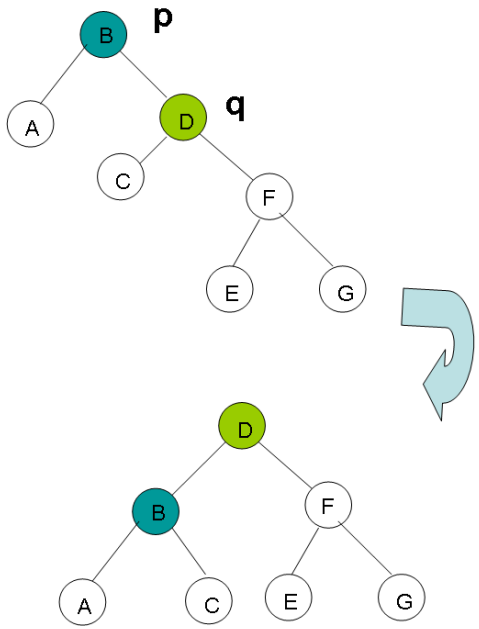


– Percurso em *inordem*: *A B C D E F G*

- Implementação:

```
/* recebe no p e rotaciona filho direito
    q. Retorna nó rotacionado */

void RotEsq (NoArvBin *p, NoArvBin **q) {
  *q = p->dir ;
   p->dir = (*q)->esq ;
   (*q)->esq = p ;
}
```

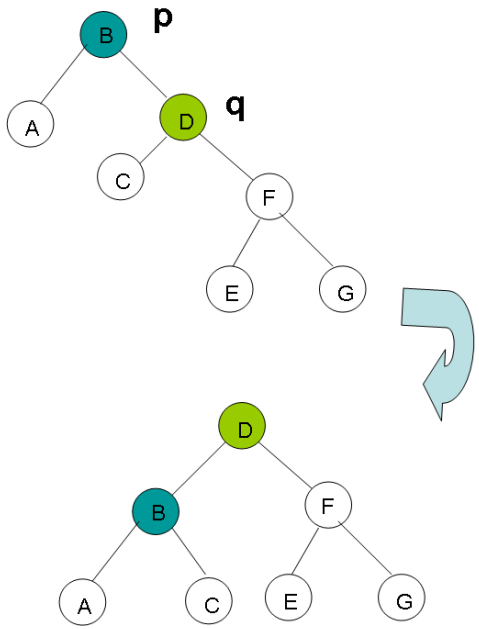


– Exercício: Escrever funções de rotação que recebam um nó e o rotacione à esquerda ou à direita de seu pai.

- Implementação:

```
/* recebe no p e rotaciona filho direito
    q. Retorna nó rotacionado */

void RotEsq (NoArvBin *p, NoArvBin **q) {
  *q = p->dir ;
   p->dir = (*q)->esq ;
   (*q)->esq = p ;
}
```



– Exercício: Escrever funções de rotação que recebam um nó e o rotacione à esquerda ou à direita de seu pai.

Árvores auto-ajustadas

1. Define uma árvore com níveis de prioridade.
2. Realiza um balanceamento (ou semi-balanceamento) de uma árvore binária de busca em função da frequência de ocorrência das chaves.
3. Os elementos que aparecem com alta frequência devem ficar em níveis superiores da árvore (possuem maior probabilidade de ocorrência).
4. Os nós podem conter um campo que indica o número de acesso ou aparição de uma chave em qualquer operação, e a árvore é varrida movendo-se o respectivo nó na direção da raiz.

Árvores auto-ajustadas

1. Define uma árvore com níveis de prioridade.
2. Realiza um balanceamento (ou semi-balanceamento) de uma árvore binária de busca em função da frequência de ocorrência das chaves.
3. Os elementos que aparecem com alta frequência devem ficar em níveis superiores da árvore (possuem maior probabilidade de ocorrência).
4. Os nós podem conter um campo que indica o número de acesso ou aparição de uma chave em qualquer operação, e a árvore é varrida movendo-se o respectivo nó na direção da raiz.

Árvores auto-ajustadas

1. Define uma árvore com níveis de prioridade.
2. Realiza um balanceamento (ou semi-balanceamento) de uma árvore binária de busca em função da frequência de ocorrência das chaves.
3. Os elementos que aparecem com alta frequência devem ficar em níveis superiores da árvore (possuem maior probabilidade de ocorrência).
4. Os nós podem conter um campo que indica o número de acesso ou aparição de uma chave em qualquer operação, e a árvore é varrida movendo-se o respectivo nó na direção da raiz.

Árvores auto-ajustadas

1. Define uma árvore com níveis de prioridade.
2. Realiza um balanceamento (ou semi-balanceamento) de uma árvore binária de busca em função da frequência de ocorrência das chaves.
3. Os elementos que aparecem com alta frequência devem ficar em níveis superiores da árvore (possuem maior probabilidade de ocorrência).
4. Os nós podem conter um campo que indica o número de acesso ou aparição de uma chave em qualquer operação, e a árvore é varrida movendo-se o respectivo nó na direção da raiz.

Exemplos de técnicas

- **Auto-reestruturação simples**

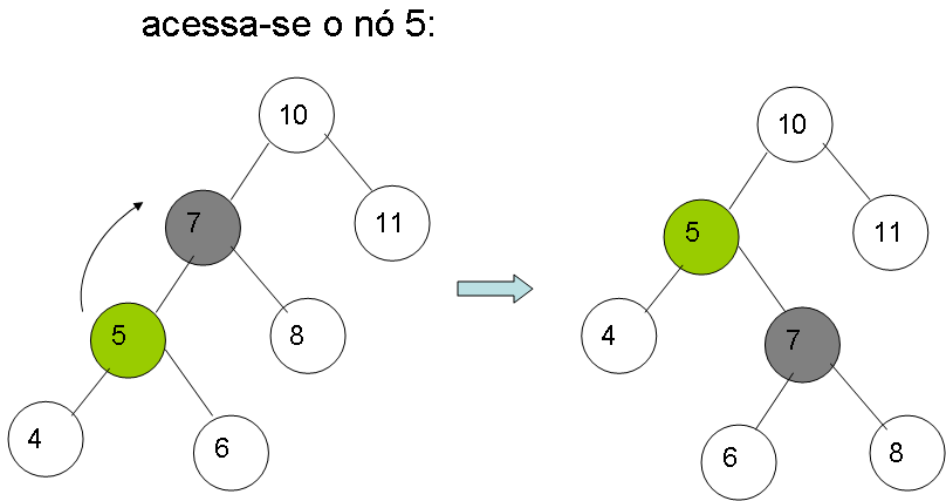
– Considera duas possibilidades:

  - Rotação simples: gira um filho ao redor de seu ascendente a cada acesso ou ocorrência do elemento filho.

- **Auto-reestruturação simples**

– Considera duas possibilidades:

  - Rotação simples: gira um filho ao redor de seu ascendente a cada acesso ou ocorrência do elemento filho.

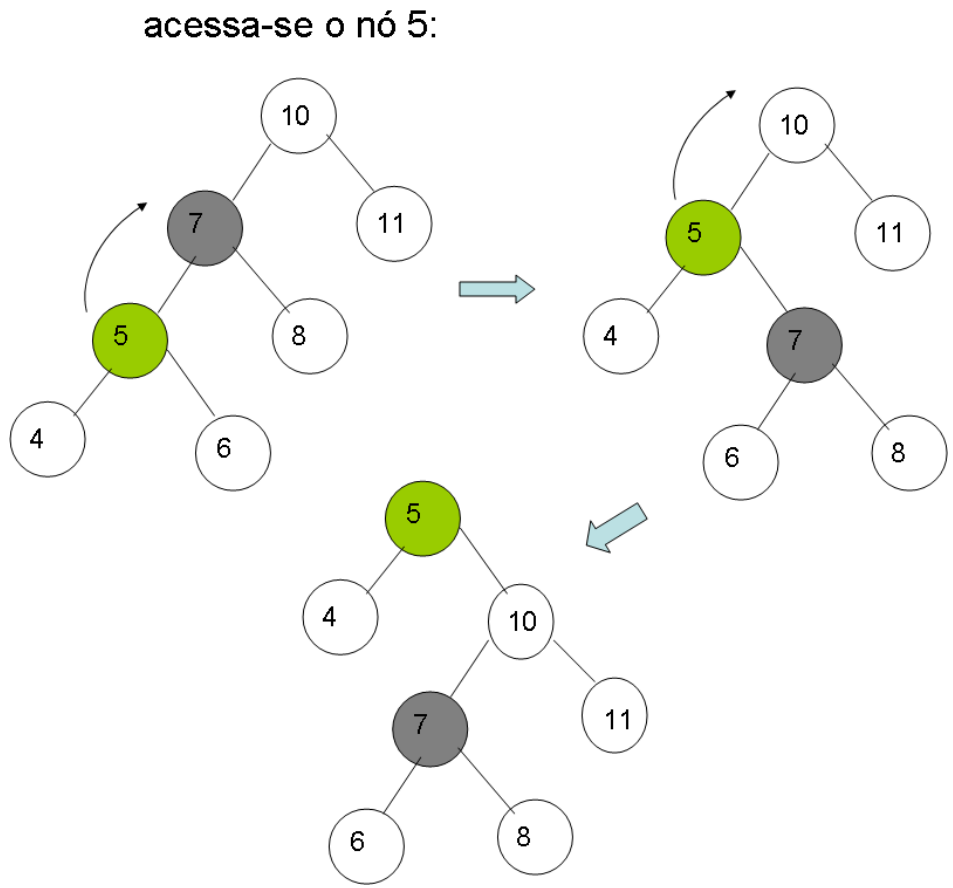


Exemplos de técnicas

- Rotação até a raiz: repete-se a rotação simples até que o elemento acessado atinja a raiz (presume-se, neste caso, alta probabilidade de ocorrência do elemento).

Exemplos de técnicas

- Rotação até a raiz: repete-se a rotação simples até que o elemento acessado atinja a raiz (presume-se, neste caso, alta probabilidade de ocorrência do elemento).



Exemplos de técnicas

- **Afunilamento**

– Aplica-se rotações simples em pares, numa ordem que depende dos vínculos entre o *filho*, o *pai* e o *avô*.

– Considera três casos distintos:
Seja R um nó sendo acessado, Q seu pai e P seu avô (se houver):

1. Caso 1: O nó ascendente de R é a raiz.
2. Caso 2: Configuração homogênea. O nó R é o filho à esquerda de seu ascendente Q; Q é o filho à esquerda de seu ascendente P, ou R e Q são filhos à direita.
3. : Caso 3: Configuração heterogênea. O nó R é o filho à direita de seu ascendente Q; Q é o filho à esquerda de seu ascendente P, ou R é o filho à esquerda de Q e Q é o filho à direita de P.

Exemplos de técnicas

- **Afunilamento**

– Aplica-se rotações simples em pares, numa ordem que depende dos vínculos entre o *filho*, o *pai* e o *avô*.

– Considera três casos distintos:
Seja R um nó sendo acessado, Q seu pai e P seu avô (se houver):

1. Caso 1: O nó ascendente de R é a raiz.
2. Caso 2: Configuração homogênea. O nó R é o filho à esquerda de seu ascendente Q; Q é o filho à esquerda de seu ascendente P, ou R e Q são filhos à direita.
3. : Caso 3: Configuração heterogênea. O nó R é o filho à direita de seu ascendente Q; Q é o filho à esquerda de seu ascendente P, ou R é o filho à esquerda de Q e Q é o filho à direita de P.

Exemplos de técnicas

- **Afunilamento**

– Aplica-se rotações simples em pares, numa ordem que depende dos vínculos entre o *filho*, o *pai* e o *avô*.

– Considera três casos distintos:
Seja R um nó sendo acessado, Q seu pai e P seu avô (se houver):

1. Caso 1: O nó ascendente de R é a raiz.
2. Caso 2: Configuração homogênea. O nó R é o filho à esquerda de seu ascendente Q; Q é o filho à esquerda de seu ascendente P, ou R e Q são filhos à direita.
3. : Caso 3: Configuração heterogênea. O nó R é o filho à direita de seu ascendente Q; Q é o filho à esquerda de seu ascendente P, ou R é o filho à esquerda de Q e Q é o filho à direita de P.

Exemplos de técnicas

- **Afunilamento**

– Aplica-se rotações simples em pares, numa ordem que depende dos vínculos entre o *filho*, o *pai* e o *avô*.

– Considera três casos distintos:
Seja R um nó sendo acessado, Q seu pai e P seu avô (se houver):

1. Caso 1: O nó ascendente de R é a raiz.
2. Caso 2: Configuração homogênea. O nó R é o filho à esquerda de seu ascendente Q; Q é o filho à esquerda de seu ascendente P, ou R e Q são filhos à direita.
3. : Caso 3: Configuração heterogênea. O nó R é o filho à direita de seu ascendente Q; Q é o filho à esquerda de seu ascendente P, ou R é o filho à esquerda de Q e Q é o filho à direita de P.

Exemplos de técnicas

- **Afunilamento**

– Aplica-se rotações simples em pares, numa ordem que depende dos vínculos entre o *filho*, o *pai* e o *avô*.

– Considera três casos distintos:
Seja R um nó sendo acessado, Q seu pai e P seu avô (se houver):

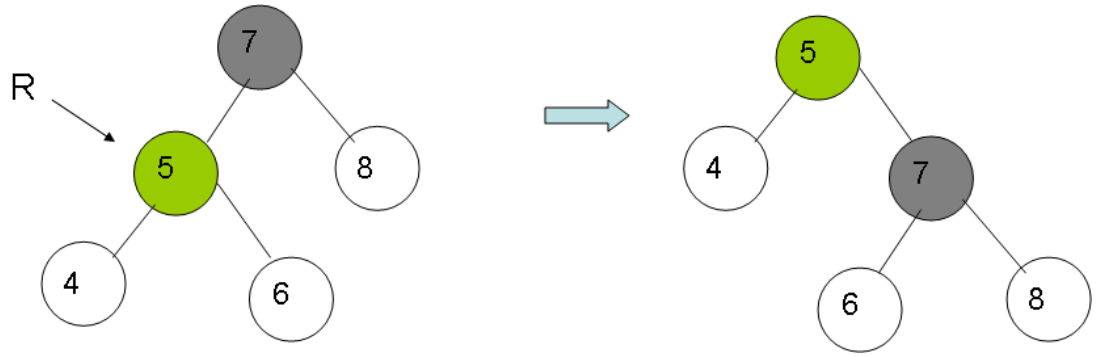
1. Caso 1: O nó ascendente de R é a raiz.
2. Caso 2: Configuração homogênea. O nó R é o filho à esquerda de seu ascendente Q; Q é o filho à esquerda de seu ascendente P, ou R e Q são filhos à direita.
3. : Caso 3: Configuração heterogênea. O nó R é o filho à direita de seu ascendente Q; Q é o filho à esquerda de seu ascendente P, ou R é o filho à esquerda de Q e Q é o filho à direita de P.

- O algoritmo para mover um nó R acessado para a raiz é:

```
Afunilamento (P, Q, R)
 - Enquanto R não é a raiz:
 |   se o ascendente de R é a raiz:
 |     afunilamento simples: gire R ao redor de seu
 |     ascendente.
 |   else se R tem configuração homogênea com seus
 |   ascendentes:
 |      afunilamento homogêneo: gire Q ao redor de P e
 |      depois R ao redor de Q.
 |   else  /* configuração heterogênea */
 |      afunilamento heterogêneo: gire R ao redor de Q
 |      e depois ao redor de P
 - Fim_Enquanto
```

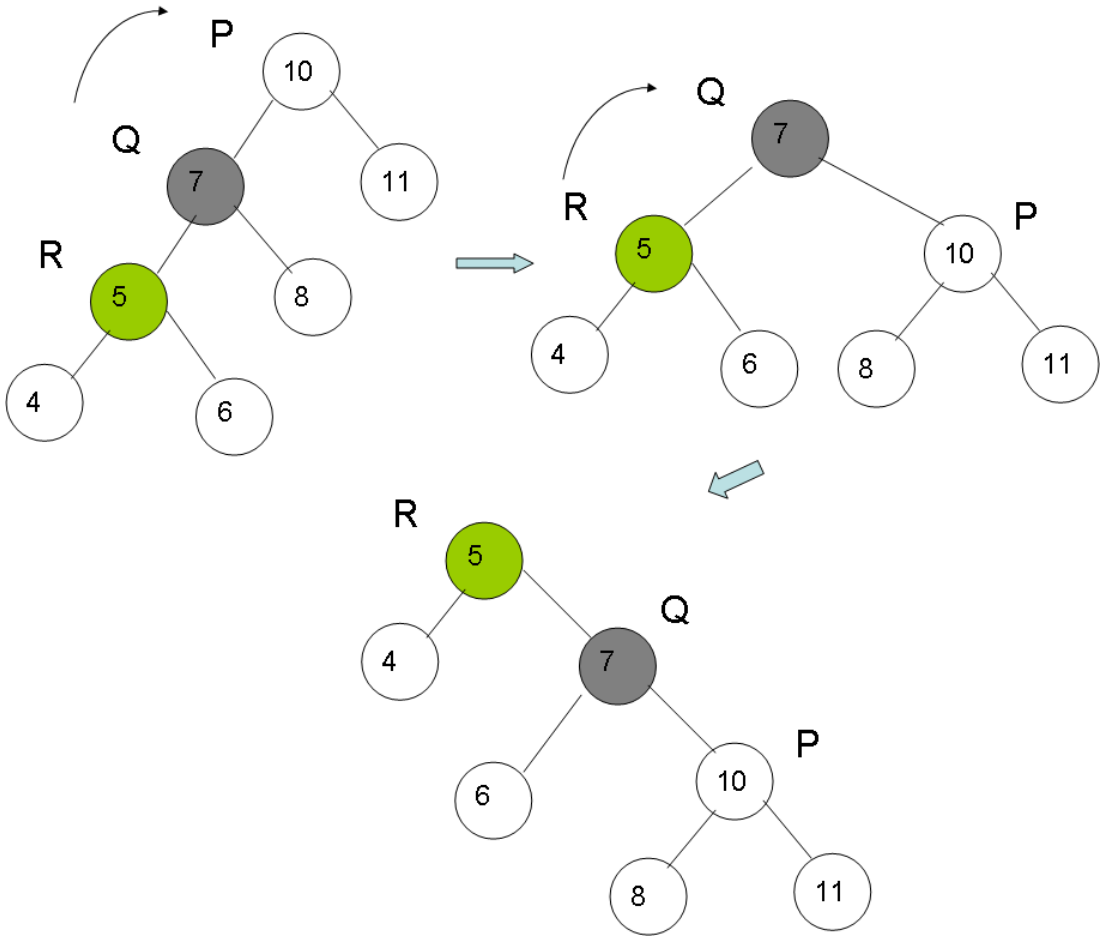
- Exemplo dos casos:

– Caso 1:



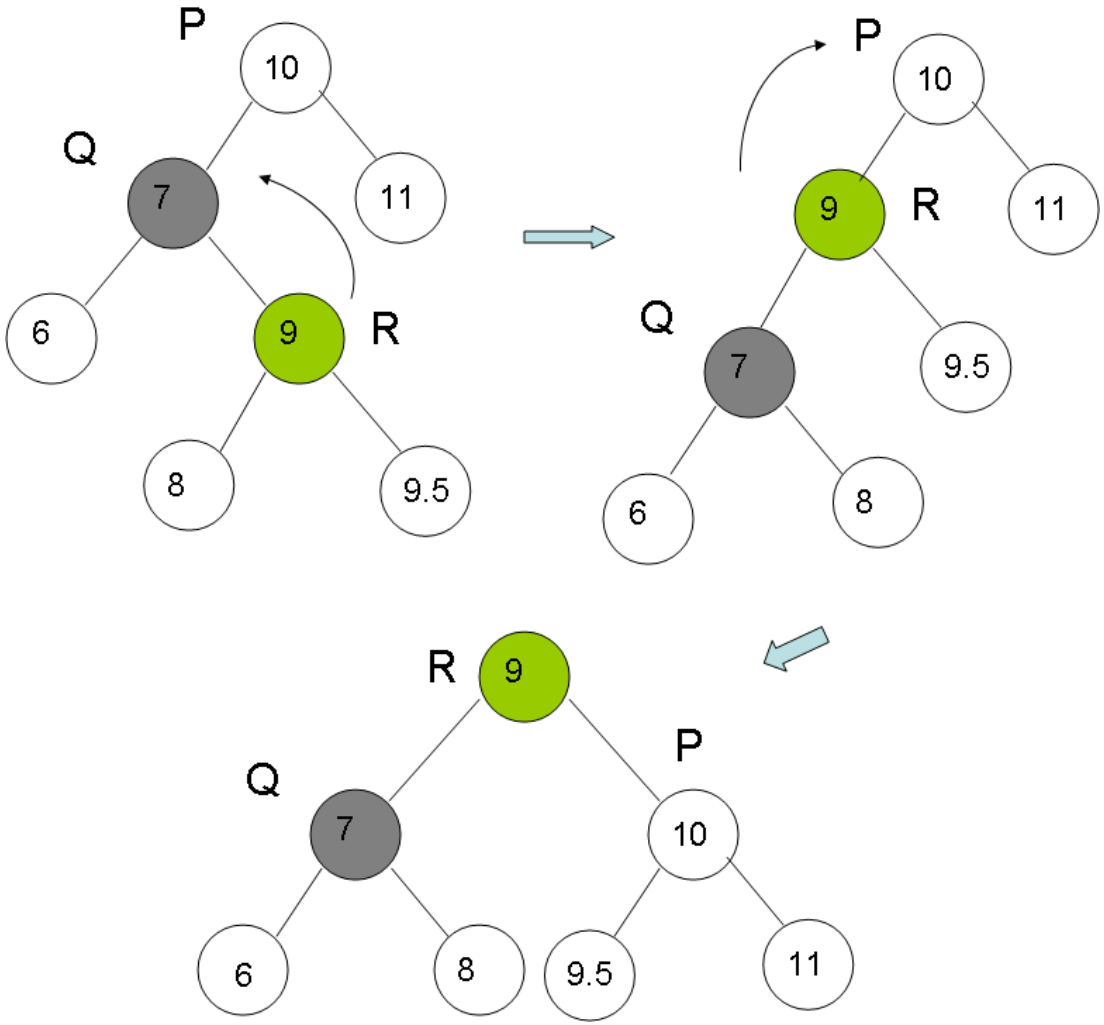
- Exemplo dos casos:

– Caso 2:



- Exemplo dos casos:

– Caso 3:

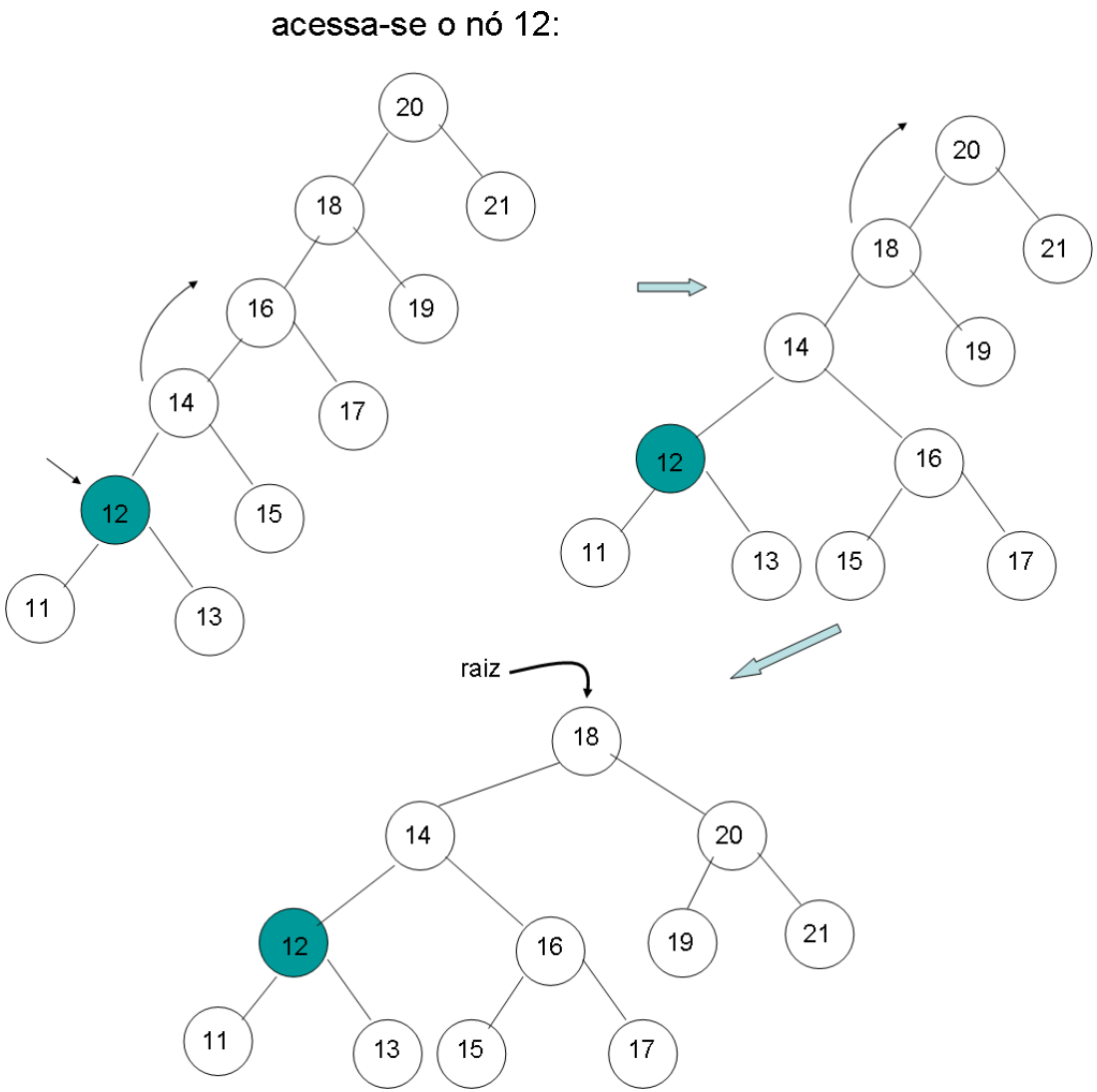


Exemplos de técnicas

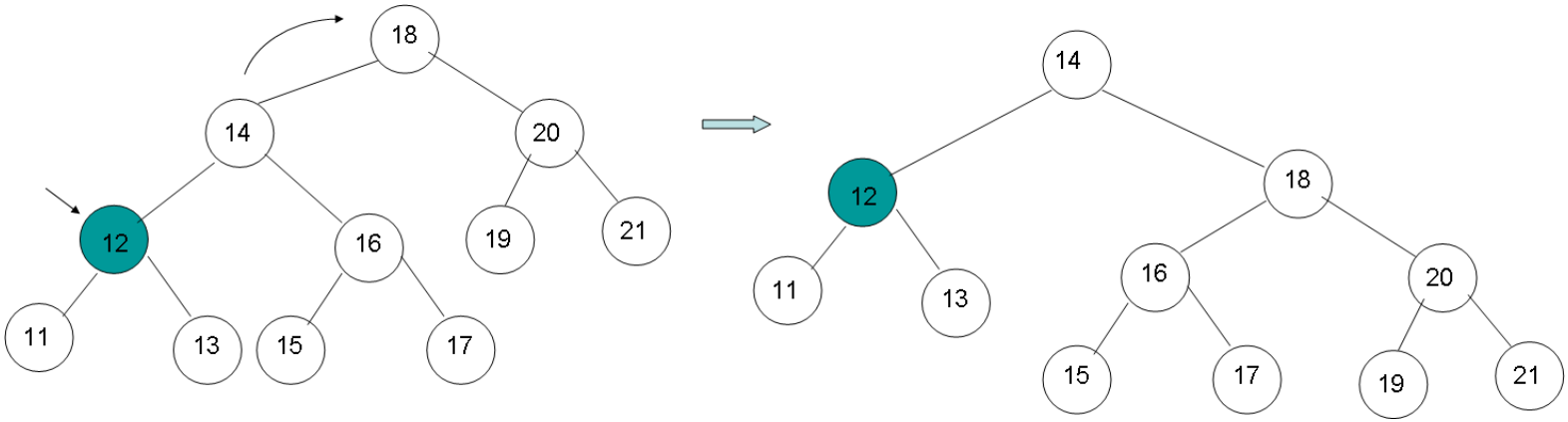
- **Semi-afunilamento**

– Executa uma rotação simples do ascendente do nó correntemente considerado, de acordo com algum critério de execução (por exemplo, até se atingir a raiz).

- Exemplo:


acessa-se o nó 12:
20
18
21
16
19
14
17
12
15
11
13
20
18
21
14
19
12
16
11
13
15
17
raiz
18
14
20
12
16
19
21
11
13
15
17

- Exemplo: acessando-se o nó 12 mais uma vez.



- Exemplo 2: Acessando-se o nó 12.