

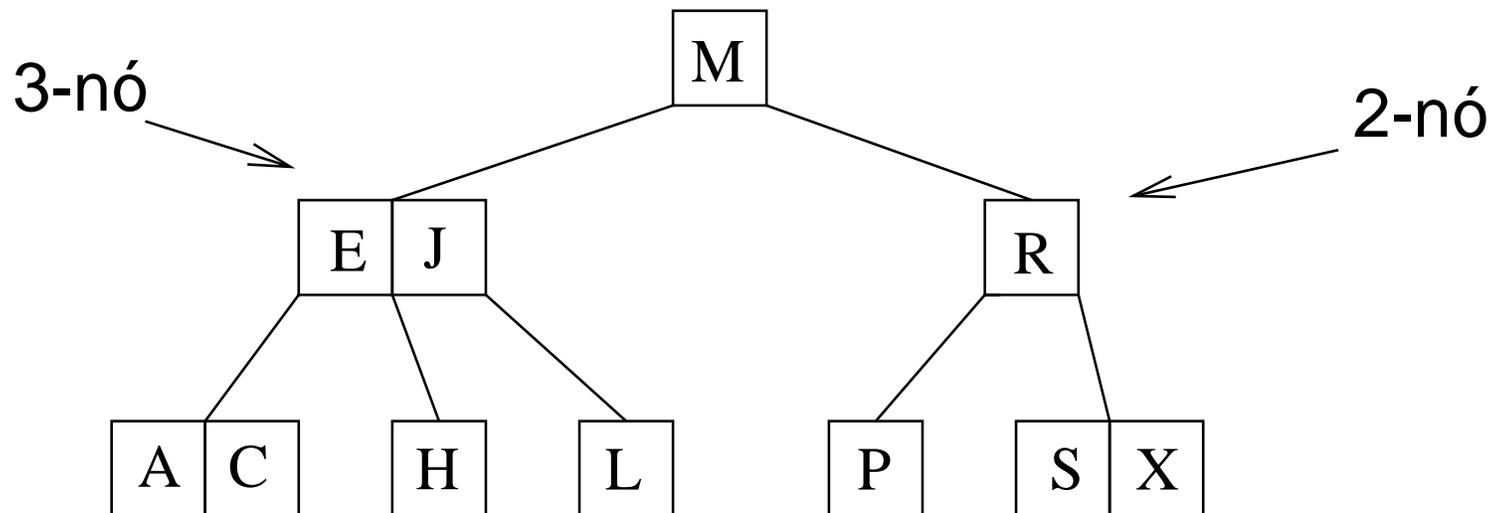
Estruturas de Dados

Cristina Gomes Fernandes

Árvores balanceadas

Uma **árvore 2-3** é uma árvore vazia ou

- um **2-nó**, com uma chave e duas subárvores 2-3;
- um **3-nó**, com duas chaves e três subárvores 2-3;
- todos os apontadores `NIL` no mesmo nível.

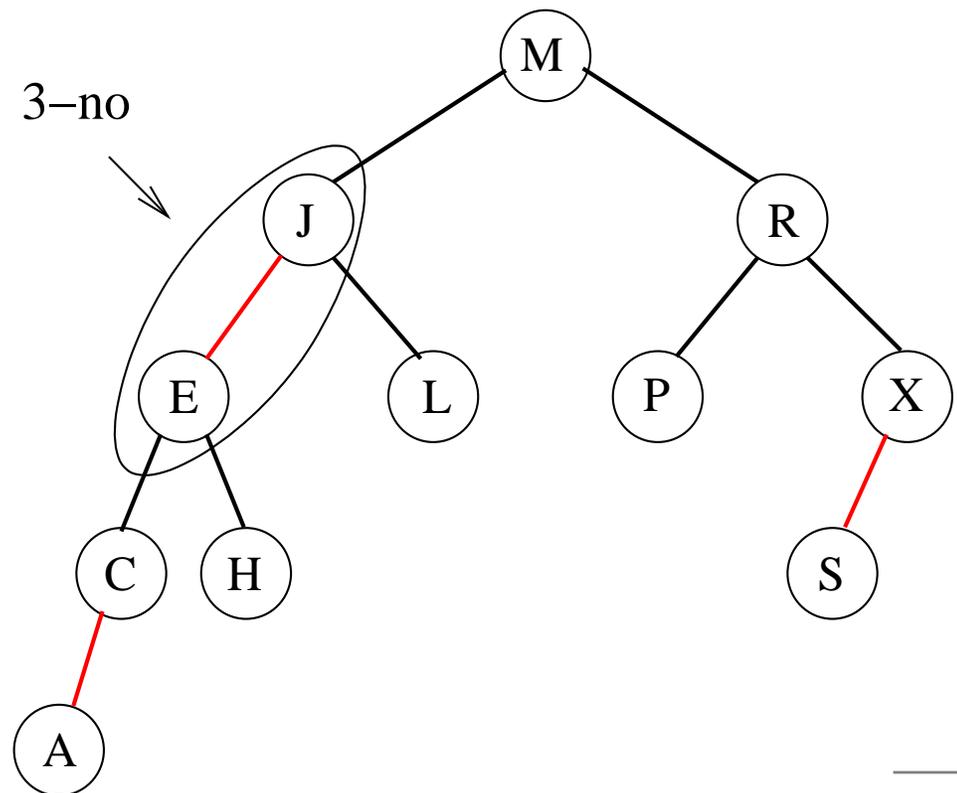
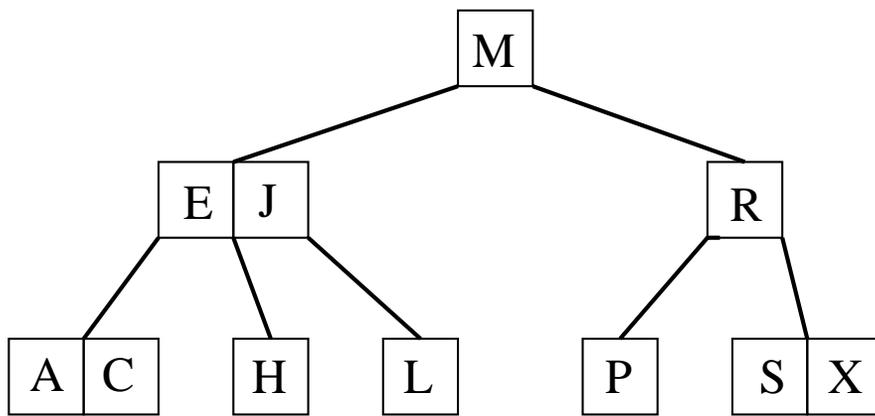


Sua altura é $O(\lg n)$, onde n é o seu número de nós.

Árvores 2-3 e ABBs rubro-negras

Árvores **rubro-negras** são árvores binárias que representam árvores 2-3.

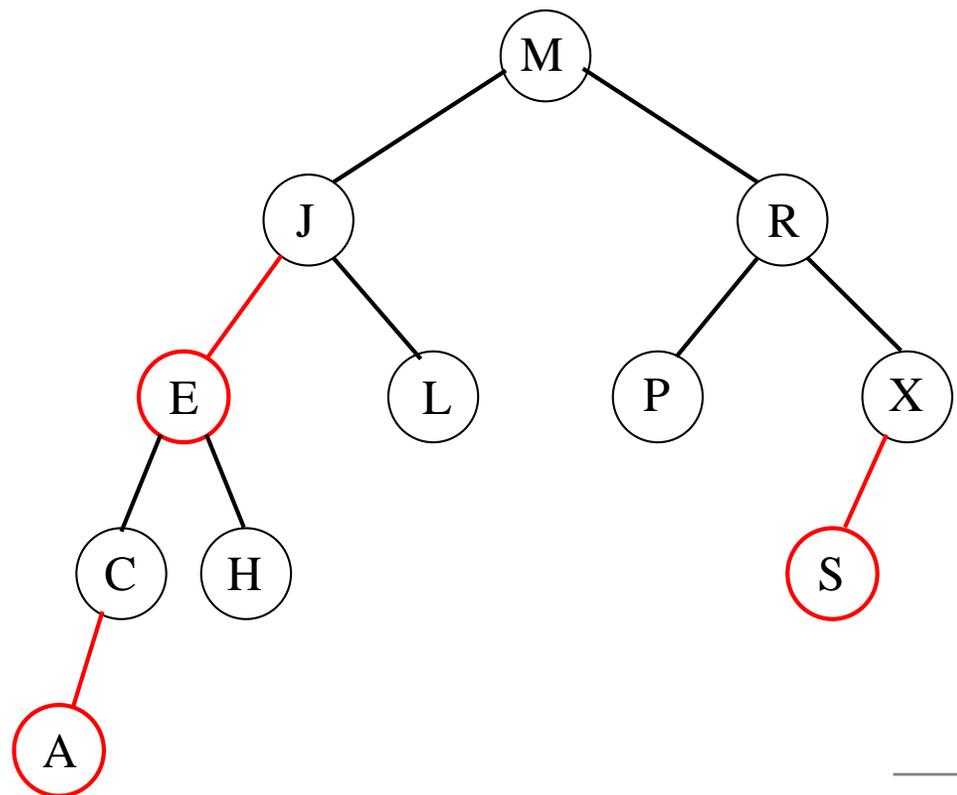
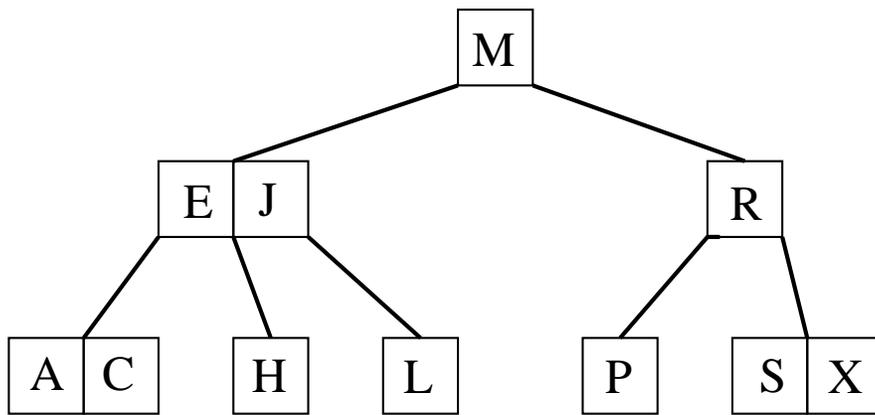
3-nós são representados por:
dois 2-nós, ligados por uma aresta vermelha.



Árvores 2-3 e ABBs rubro-negras

Árvores **rubro-negras** são árvores binárias que representam árvores 2-3.

3-nós são representados por:
dois 2-nós, ligados por uma aresta vermelha.



Nós são **rubros** ou **negros**.

ABBs rubro-negras

Uma ABB é **rubro-negra** se

1. todo nó é **rubro** ou **negro**
2. toda folha (NIL) é **negra**
3. se um nó é **rubro**,
então é filho esquerdo de um nó **negro**
4. todo caminho de um nó x até uma folha sua descendente tem o mesmo número de nós **negros**.

ABBs rubro-negras

Uma ABB é **rubro-negra** se

1. todo nó é **rubro** ou **negro**
2. toda folha (NIL) é **negra**
3. se um nó é **rubro**,
então é filho esquerdo de um nó **negro**
4. todo caminho de um nó x até uma folha sua descendente tem o mesmo número de nós **negros**.

Altura negra de x : número de nós **negros** no caminho de um filho de x até uma folha descendente de x .

Lema: Uma ABB **rubro-negra** com n nós internos tem altura no máximo $2 \lg(n + 1)$.

Inserção em ABB

INSIRA (T, x)

1 $T \leftarrow \text{INSIRAREC}(T, x)$

INSIRAREC (T, x)

1 **se** $T = \text{NIL}$

2 **então** $q \leftarrow \text{NOVACÉLULA}(x, \text{NIL}, \text{NIL})$

3 **devolva** q

4 **se** $x < \text{info}(T)$

5 **então** $\text{esq}(T) \leftarrow \text{INSIRAREC}(\text{esq}(T), x)$

6 **senão** $\text{dir}(T) \leftarrow \text{INSIRAREC}(\text{dir}(T), x)$

7 **devolva** T

Inserção em ABB rubro-negra

INSIRA (T, x)

- 1 $T \leftarrow$ **INSIRAREC**(T, x)
- 2 $cor(T) \leftarrow$ NEGRO ▷ linha nova!

INSIRAREC (T, x)

- 1 **se** $T = \text{NIL}$
- 2 **então** $q \leftarrow$ NOVACÉLULA($x, \text{NIL}, \text{NIL}, \text{RUBRO}$)
- 3 **devolva** q
- 4 **se** $x < info(T)$
- 5 **então** $esq(T) \leftarrow$ **INSIRAREC**($esq(T), x$)
- 6 **senão** $dir(T) \leftarrow$ **INSIRAREC**($dir(T), x$)
- 7 **BALANCEIE**(T) ▷ linha nova!
- 8 **devolva** T

Rotinas auxiliares

BALANCEIE (T)

- 1 **se** NEGRO($esq(T)$) **e** RUBRO($dir(T)$)
- 2 **então** GIREESQ(T)
- 3 **se** RUBRO($esq(T)$) **e** RUBRO($esq(esq(T))$)
- 4 **então** $T \leftarrow$ GIREDIR(T)
- 5 **se** RUBRO($esq(T)$) **e** RUBRO($dir(T)$)
- 6 **então** TROQUECORES(T)
- 7 devolva T

NEGRO (p)

- 1 **se** $p = \text{NIL}$ **ou** $cor(p) = \text{NEGRO}$
- 2 **então devolva** VERDADE
- 3 **senão devolva** FALSO

RUBRO (p)

- 1 **devolva** não NEGRO (p)

rotações

O nó p é tal que $dir(p) \neq \text{NIL}$ e $cor(dir(p)) = \text{RUBRO}$.

GIREESQ (p)

- 1 $q \leftarrow dir(p)$
- 2 $dir(p) \leftarrow esq(q)$
- 3 $esq(q) \leftarrow p$
- 4 $cor(q) \leftarrow cor(p)$
- 5 $cor(p) \leftarrow \text{RUBRO}$
- 6 **devolva** q

O nó p é interno.

O **GIREDIR** (p) é semelhante.

Troca de cores

OUTRACOR (c)

- 1 **se** $c = \text{RUBRO}$
- 2 **então devolva** NEGRO
- 3 **senão devolva** RUBRO

TROQUECORES (p)

- 1 $cor(p) \leftarrow \text{OUTRACOR}(cor(p))$
- 2 $cor(esq(p)) \leftarrow \text{OUTRACOR}(cor(esq(p)))$
- 3 $cor(dir(p)) \leftarrow \text{OUTRACOR}(cor(dir(p)))$

Remoção em árvore 2-3

Como na ABB,
se a chave a ser removida está em um nó interno,
a trocamos pela menor da subárvore a sua direita.

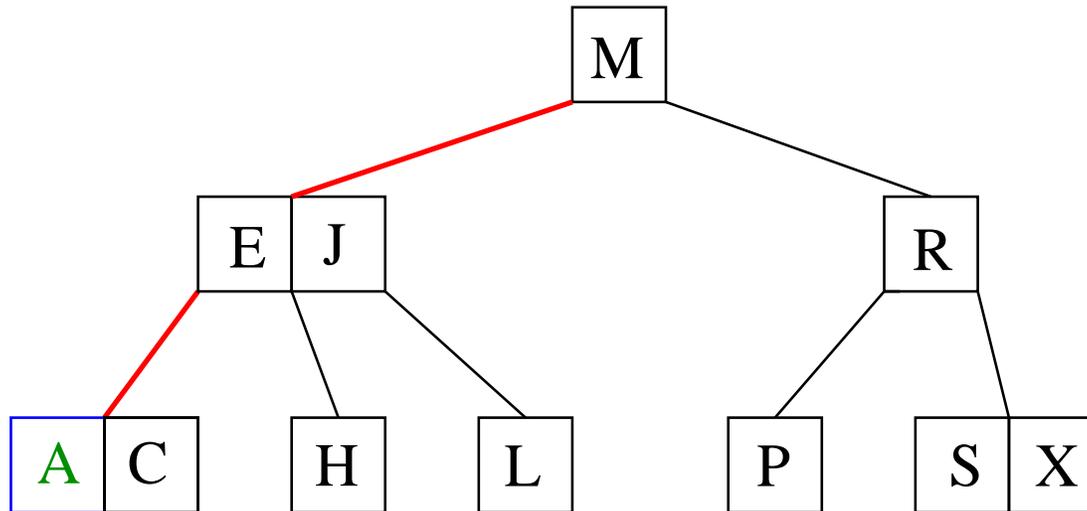
Remoção em árvore 2-3

Como na ABB,
se a chave a ser removida está em um nó interno,
a trocamos pela menor da subárvore a sua direita.

Assim podemos nos concentrar em remoções de chaves
das folhas. Na verdade, podemos nos concentrar na
remoção do mínimo de uma árvore 2-3.

Remoção em árvore 2-3

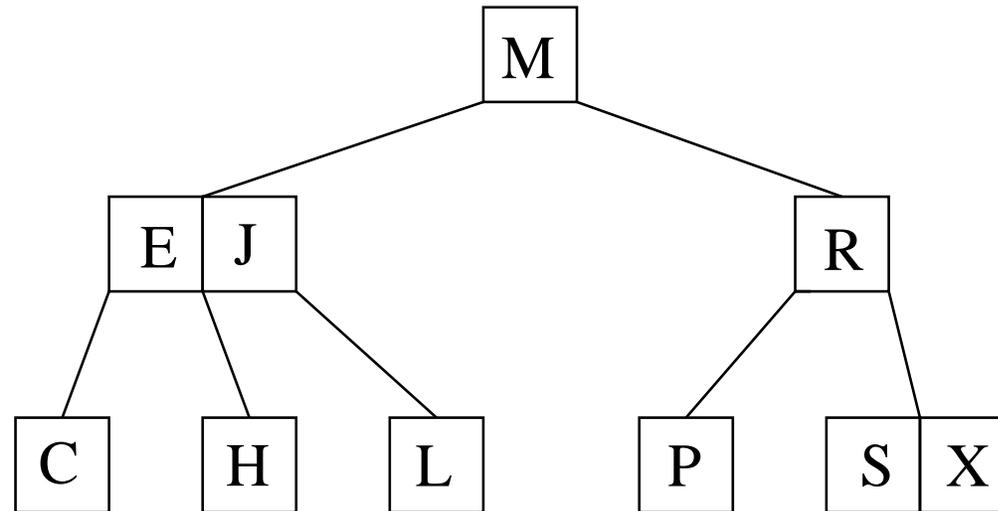
Remoção do A.



Busca pelo A.

Remoção em árvore 2-3

Remoção do A.

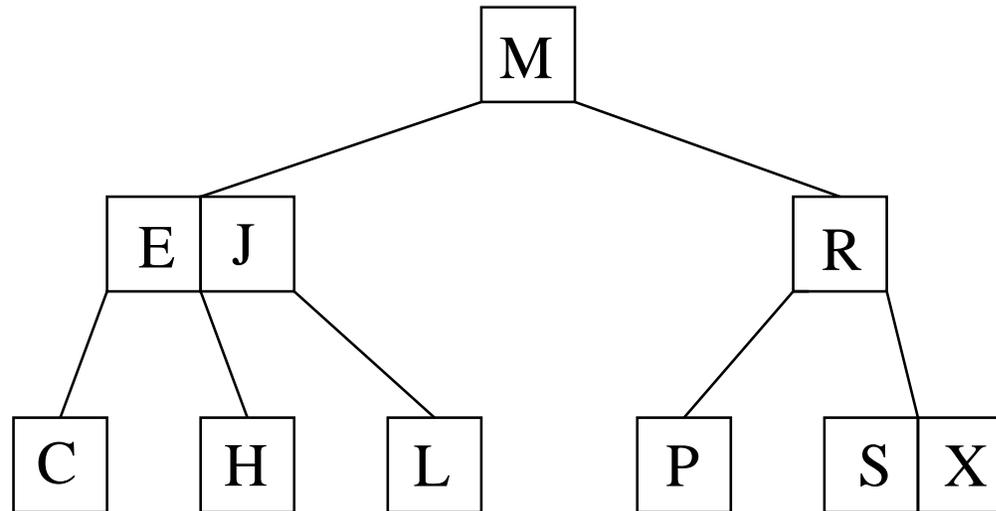


Busca pelo A.

Se a busca termina em um 3-nó,
remove A neste nó.

Remoção em árvore 2-3

Remoção do A.



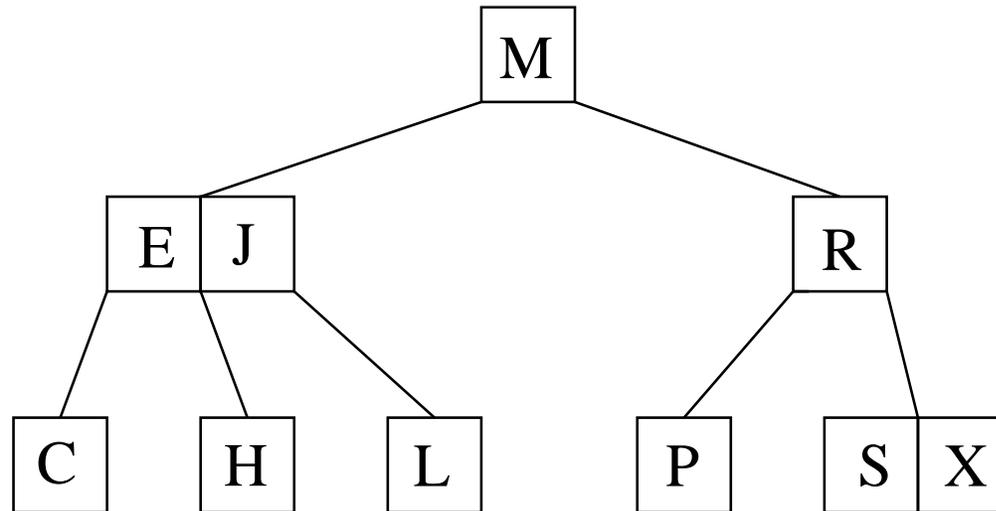
Busca pelo A.

Se a busca termina em um 3-nó,
remove A neste nó.

E se a busca termina num 2-nó?

Remoção em árvore 2-3

Remoção do A.



Busca pelo A.

Se a busca termina em um 3-nó,
remove A neste nó.

E se a busca termina num 2-nó?

Evita-se esse caso!

Remoção em árvore 2-3

E se a busca termina num 2-nó? **Evita-se esse caso!**

Como???

Remoção em árvore 2-3

E se a busca termina num 2-nó? **Evita-se esse caso!**

Como???

Na descida da busca, altera-se a árvore 2-3, de modo que, ao chegarmos na folha, ela seja um 3-nó.

Remoção em árvore 2-3

E se a busca termina num 2-nó? **Evita-se esse caso!**

Como???

Na descida da busca, altera-se a árvore 2-3, de modo que, ao chegarmos na folha, ela seja um 3-nó.

Na verdade, a busca é feita sempre garantindo que a raiz da árvore é um 3-nó ou um 4-nó temporário.

Na volta da recursão, eventuais 4-nós são desfeitos.

Remoção em ABBs

REMOVA (T, q)

```
1  se  $dir(q) = \text{NIL}$ 
2    então  $p \leftarrow q$ 
3           $f \leftarrow esq(p)$ 
4    senão  $p \leftarrow \text{MÍNIMO}(dir(q))$ 
5           $info(q) \leftarrow info(p)$ 
6           $f \leftarrow dir(p)$ 
7  se  $f \neq \text{NIL}$ 
8    então  $pai(f) \leftarrow pai(p)$ 
9  se  $pai(p) = \text{NIL}$ 
10   então  $T \leftarrow f$ 
11   senão se  $p = esq(pai(p))$ 
12         então  $esq(pai(p)) \leftarrow f$ 
13         senão  $dir(pai(p)) \leftarrow f$ 
14  LIBERACÉLULA( $p$ )
```

Versão recursiva da remoção

Supõe que x está presente na ABB.

REMOVA (T, x)

1 $T \leftarrow \text{REMOVAREC}(T, x)$

REMOVAREC (T, x)

1 **se** $x < \text{info}(T)$

2 **então** $\text{esq}(T) \leftarrow \text{REMOVAREC}(\text{esq}(T), x)$

3 **senão se** $x > \text{info}(T)$

4 **então** $\text{dir}(T) \leftarrow \text{REMOVAREC}(\text{dir}(T), x)$

5 **senão se** $\text{dir}(T) = \text{NIL}$

6 **então devolva** $\text{esq}(T)$

7 $\text{info}(T) \leftarrow \text{MÍNIMO}(\text{dir}(T))$

8 $\text{dir}(T) \leftarrow \text{REMOVAMIN}(\text{dir}(T))$

9 **devolva** T

Remoção do mínimo em ABB

REMOVAMIN (T)

1 **se** $esq(T) = \text{NIL}$

2 **então devolva** $dir(T)$

3 **senão** $esq(T) \leftarrow \text{REMOVAMIN}(esq(T))$

4 **devolva** T

Remoção do mínimo em ABB

REMOVAMIN (T)

- 1 **se** $esq(T) = \text{NIL}$
- 2 **então devolva** $dir(T)$
- 3 **senão** $esq(T) \leftarrow \text{REMOVAMIN}(esq(T))$
- 4 **devolva** T

Exercício: Modifique as rotinas acima para que a remoção funcione mesmo que x não esteja na ABB dada.

Exercício: Ajuste estas rotinas para que mantenham o campo pai atualizado.

Remoção do mínimo em rubro-negra

A rotina abaixo recebe uma árvore rubro-negra a menos da raiz, que pode ser vermelha.

Ademais, $cor(T) = \text{RUBRO}$ ou $cor(esq(T)) = \text{RUBRO}$.

(Isso corresponde a supor que T é um 3-nó ou um 4-nó.)

Remoção do mínimo em rubro-negra

A rotina abaixo recebe uma árvore rubro-negra a menos da raiz, que pode ser vermelha.

Ademais, $cor(T) = \text{RUBRO}$ ou $cor(esq(T)) = \text{RUBRO}$.

(Isso corresponde a supor que T é um 3-nó ou um 4-nó.)

REMOVAMIN (T)

1 **se** $esq(T) = \text{NIL}$

2 **então devolva** NIL $\triangleright dir(T) = \text{NIL}$ também

3 **se** $\text{NEGRO}(esq(T))$ e $\text{NEGRO}(esq(esq(T)))$

4 **então** $T \leftarrow \text{MOVARUBROESQ}(T)$

5 $esq(T) \leftarrow \text{REMOVAMIN}(esq(T))$

6 **BALANCEIE** (T)

Rotina auxiliar para a remoção

A rotina supõe que $cor(p) = \text{RUBRO}$.

MOVARUBROESQ (p)

1 **TROQUECORES**(p)

2 **se** **RUBRO**($esq(dir(p))$)

3 **então** $dir(p) \leftarrow \text{GIREDIR}(dir(p))$

4 $p \leftarrow \text{GIREESQ}(p)$

5 **devolva** p

Rotina auxiliar para a remoção

A rotina supõe que $cor(p) = \text{RUBRO}$.

MOVARUBROESQ (p)

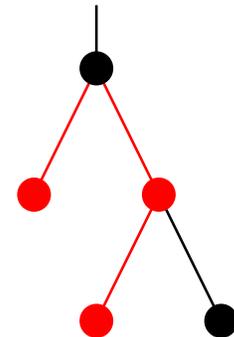
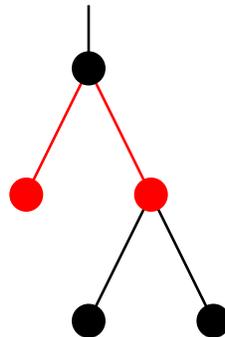
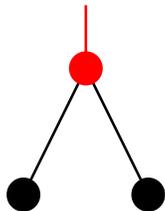
1 **TROQUECORES**(p)

2 **se** **RUBRO**($esq(dir(p))$)

3 **então** $dir(p) \leftarrow \text{GIREDIR}(dir(p))$

4 $p \leftarrow \text{GIREESQ}(p)$

5 **devolva** p



Rotina auxiliar para a remoção

A rotina supõe que $cor(p) = \text{RUBRO}$.

MOVARUBROESQ (p)

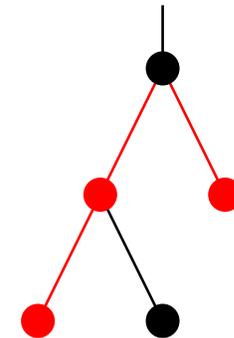
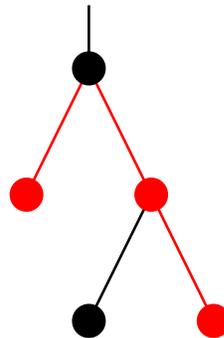
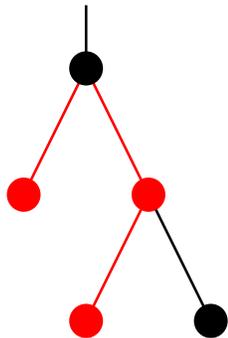
1 **TROQUECORES**(p)

2 **se** **RUBRO**($esq(dir(p))$)

3 **então** $dir(p) \leftarrow \text{GIREDIR}(dir(p))$

4 $p \leftarrow \text{GIREESQ}(p)$

5 **devolva** p



Remoção em ABB rubro-negra

A rotina supõe que $cor(p) = \text{RUBRO}$.

MOVARUBRODIR (p)

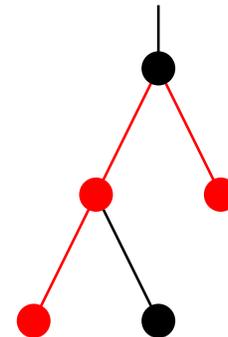
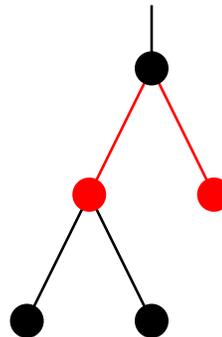
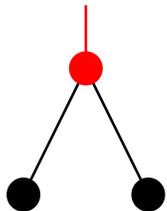
1 **TROQUECORES**(p)

2 **se** **RUBRO**($esq(esq(p))$)

3 **então** $p \leftarrow \text{GIREDIR}(p)$

4 $dir(p) \leftarrow \text{GIREESQ}(dir(p))$

5 **devolva** p



Remoção em ABB rubro-negra

A rotina supõe que $cor(p) = \text{RUBRO}$.

MOVARUBRODIR (p)

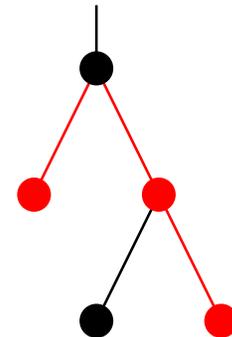
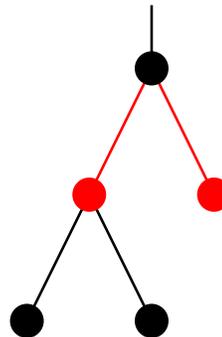
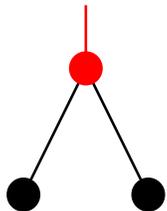
1 **TROQUECORES**(p)

2 **se** **RUBRO**($esq(esq(p))$)

3 **então** $p \leftarrow \text{GIREDIR}(p)$

4 $dir(p) \leftarrow \text{GIREESQ}(dir(p))$

5 **devolva** p



Remoção em ABB rubro-negra

A rotina supõe que $cor(p) = \text{RUBRO}$.

MOVARUBRODIR (p)

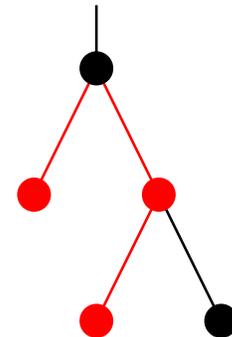
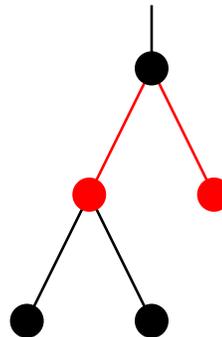
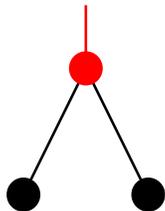
1 **TROQUECORES**(p)

2 **se** **RUBRO**($esq(esq(p))$)

3 **então** $p \leftarrow \text{GIREDIR}(p)$

4 $dir(p) \leftarrow \text{GIREESQ}(dir(p))$

5 **devolva** p



Remoção em ABB rubro-negra

A rotina supõe que $cor(p) = \text{RUBRO}$.

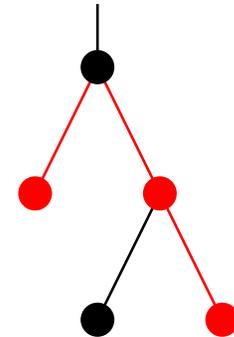
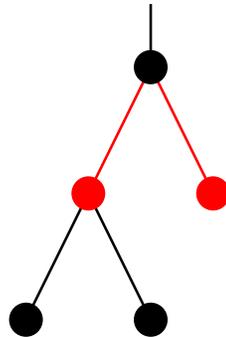
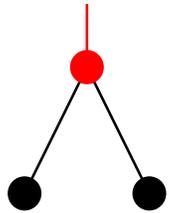
MOVARUBRODIR (p)

1 **TROQUECORES**(p)

2 **se** **RUBRO**($esq(esq(p))$)

3 **então** $p \leftarrow \text{GIREDIR}(p)$

4 **devolva** p



O **GIRESQ**($dir(p)$) é desnecessário.

Remoção em ABB rubro-negra

A rotina abaixo recebe uma árvore rubro-negra a menos da raiz, que pode ser vermelha, e o filho direito da raiz, que pode ser vermelho.

Ademais, $cor(T) = \text{RUBRO}$ ou $cor(esq(T)) = \text{RUBRO}$ ou $cor(dir(T)) = \text{RUBRO}$.

(Isso corresponde a supor que T é um 3-nó ou um 4-nó.)

REMOVA (T, x)

1 **se** NEGRO($esq(T)$) **e** NEGRO($dir(T)$)

2 **então** $cor(T) \leftarrow \text{RUBRO}$

3 $T \leftarrow \text{REMOVAREC}(T, x)$

4 **se** $T \neq \text{NIL}$

5 **então** $cor(T) \leftarrow \text{NEGRO}$ ▷ a raiz é sempre negra

Remoção em ABB rubro-negra

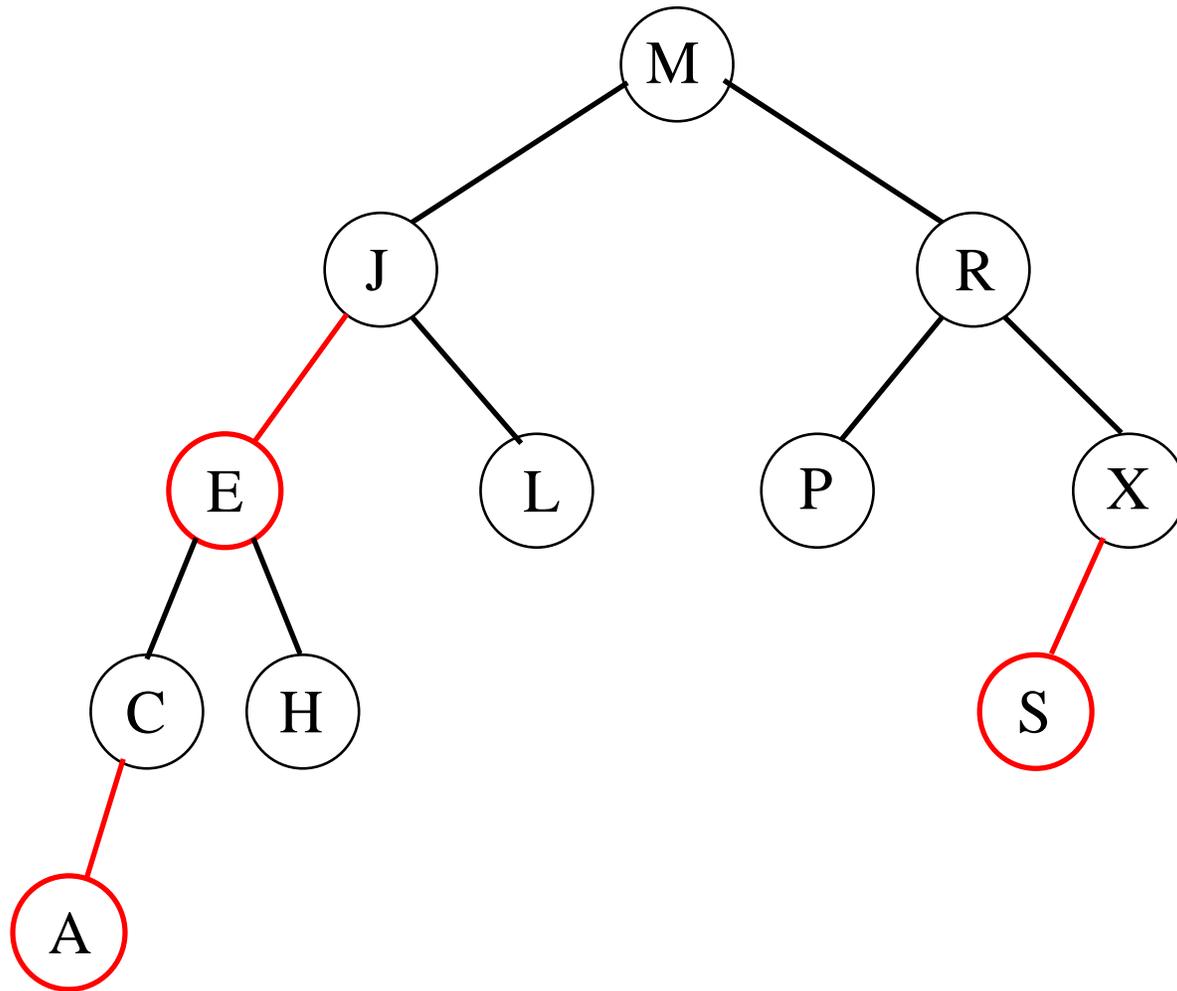
```
REMOVAREC ( $T, x$ )
1  se  $x < info(T)$ 
2    então se NEGRO( $esq(T)$ ) e NEGRO( $esq(esq(T))$ )
3      então  $T \leftarrow$  MOVARUBROESQ( $T$ )
4       $esq(T) \leftarrow$  REMOVAREC( $esq(T), x$ )
5    senão se RUBRO( $esq(T)$ )
6      então  $T \leftarrow$  GIREDIR( $T$ )
7      se  $x = info(T)$  e  $dir(T) = \text{NIL}$ 
8        então devolva NIL
9      se NEGRO( $dir(T)$ ) e NEGRO( $esq(dir(T))$ )
10     então  $T \leftarrow$  MOVARUBRODIR( $T$ )
11     se  $x = info(T)$ 
12       então  $info(T) \leftarrow$  MÍNIMO( $dir(T)$ )
13        $dir(T) \leftarrow$  REMOVAMIN( $dir(T)$ )
14     senão  $T \leftarrow$  REMOVAREC( $dir(T), x$ )
15  BALANCEIE( $T$ )
```

Remoção em ABB rubro-negra

```
REMOVAREC ( $T, x$ )
1  se  $x < \text{info}(T)$ 
2    então se NEGRO( $\text{esq}(T)$ ) e NEGRO( $\text{esq}(\text{esq}(T))$ )
3      então  $T \leftarrow \text{MOVARUBROESQ}(T)$ 
4       $\text{esq}(T) \leftarrow \text{REMOVAREC}(\text{esq}(T), x)$ 
5    senão se RUBRO( $\text{esq}(T)$ )
6      então  $T \leftarrow \text{GIREDIR}(T)$ 
7      se NEGRO( $\text{dir}(T)$ ) e NEGRO( $\text{esq}(\text{dir}(T))$ )
8        então  $T \leftarrow \text{MOVARUBRODIR}(T)$ 
9    se  $x > \text{info}(T)$ 
10     então  $\text{dir}(T) \leftarrow \text{REMOVAREC}(\text{dir}(T), x)$ 
11     senão se  $\text{dir}(T) = \text{NIL}$ 
12       então devolva NIL
13      $\text{info}(T) \leftarrow \text{MÍNIMO}(\text{dir}(T))$ 
14      $\text{dir}(T) \leftarrow \text{REMOVAMIN}(\text{dir}(T))$ 
15  BALANCEIE( $T$ )
```

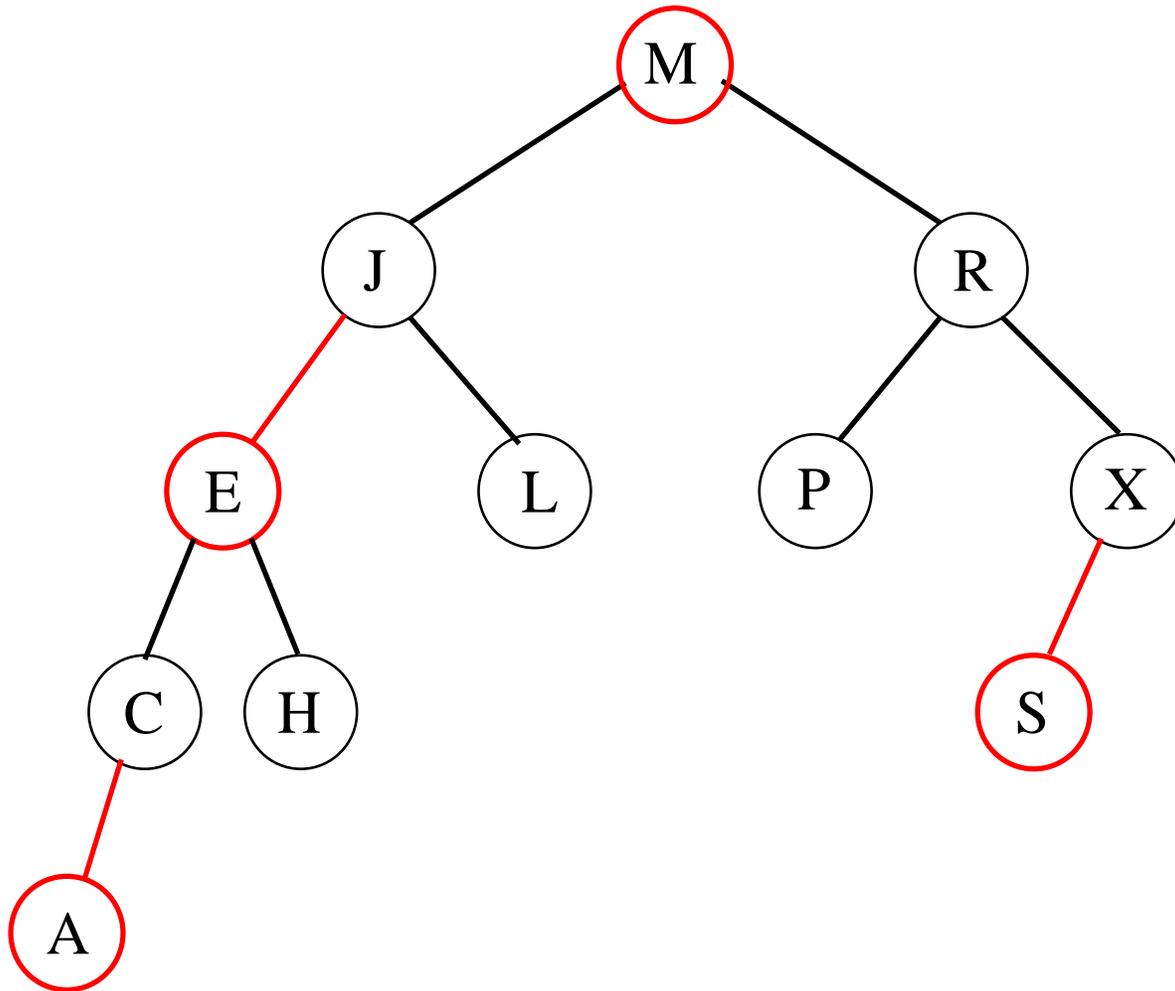
Exemplo

Remova A.



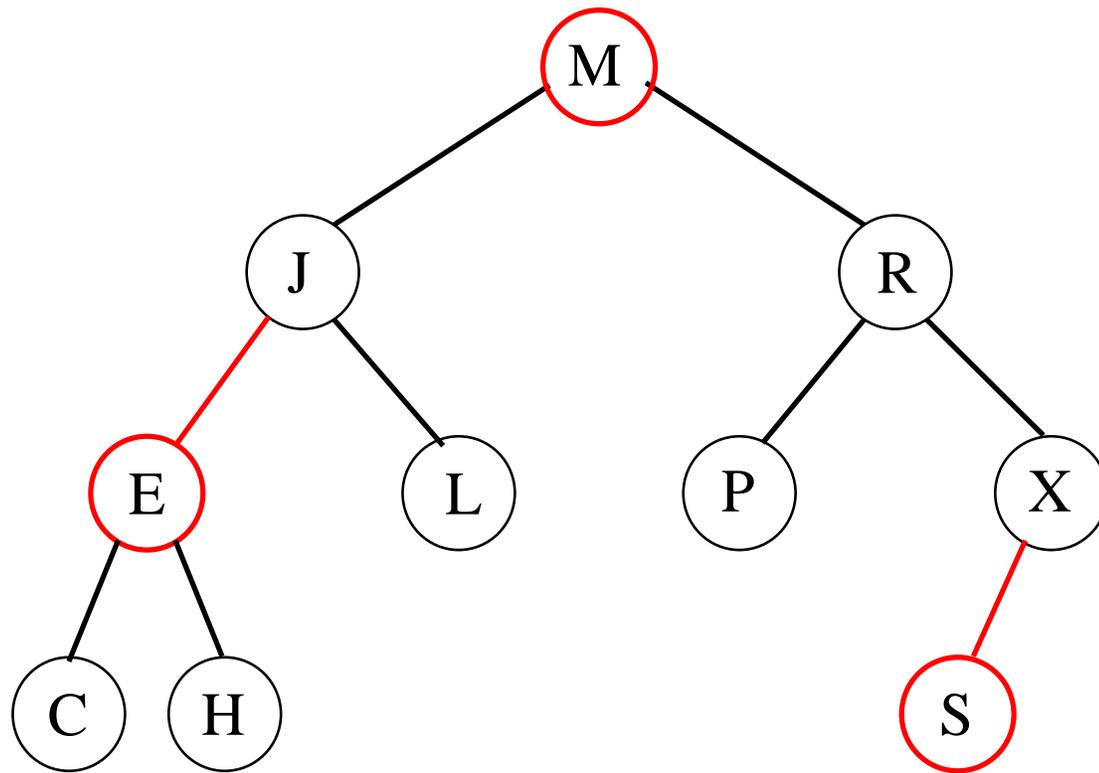
Exemplo

Remova A.



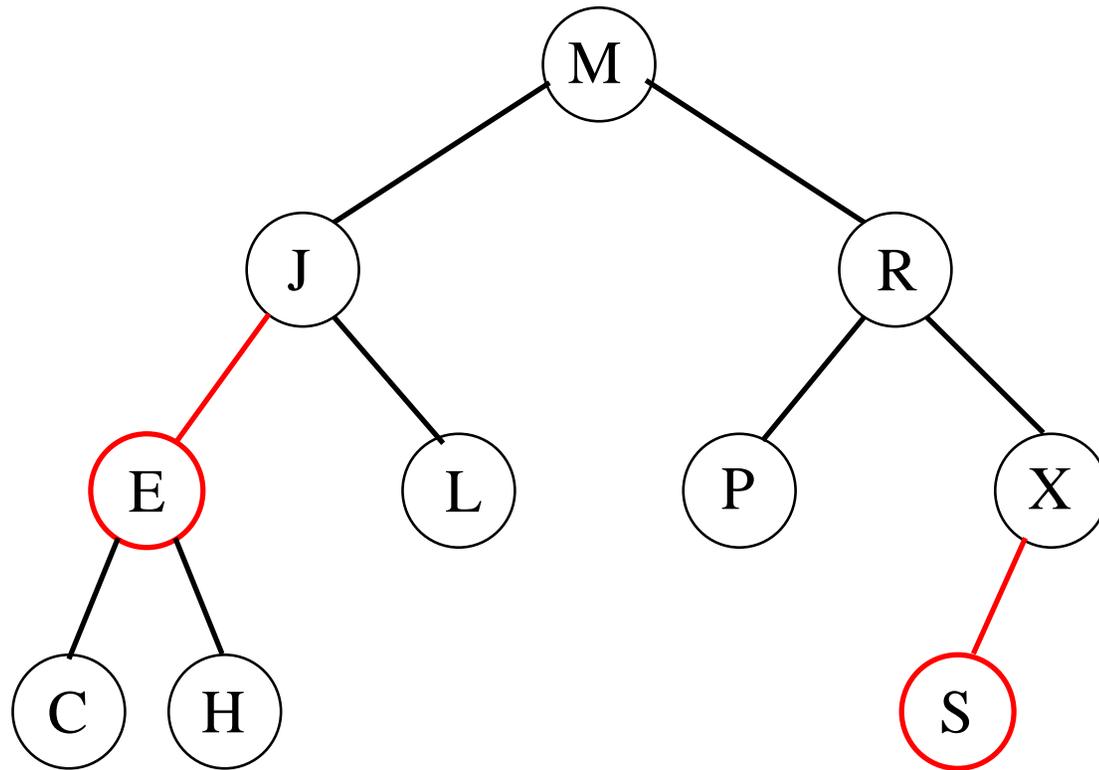
Exemplo

Remove A.



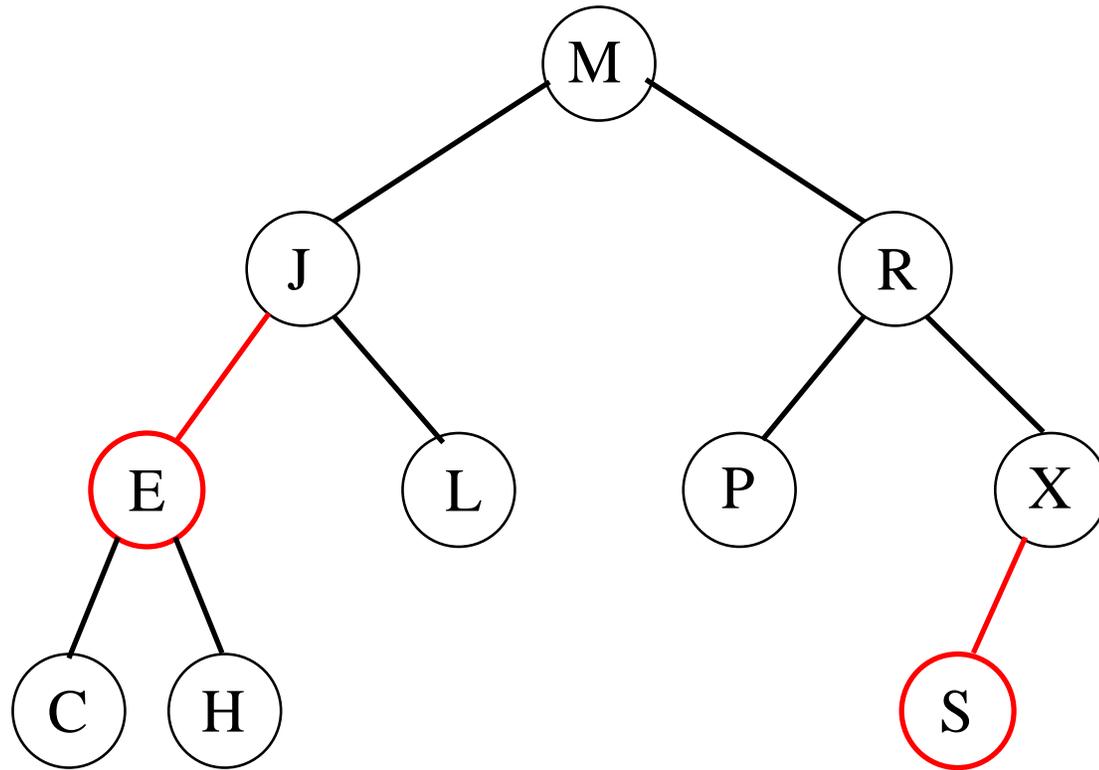
Exemplo

Remova A.



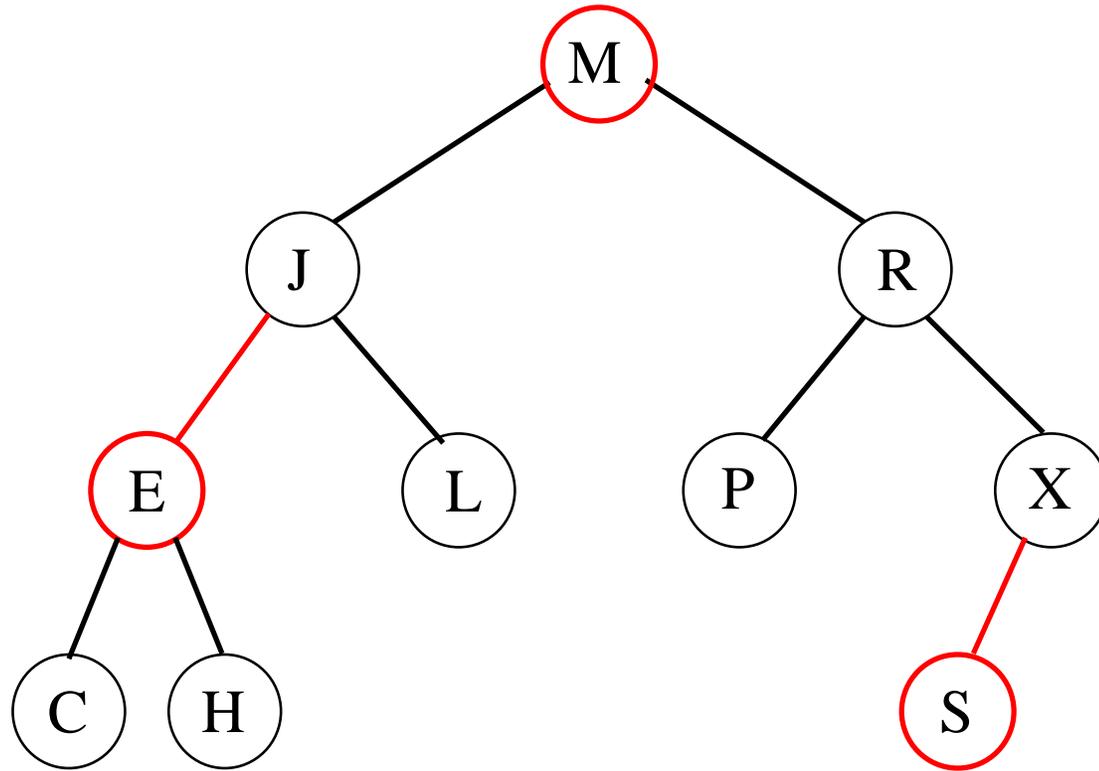
Exemplo

Remova C.



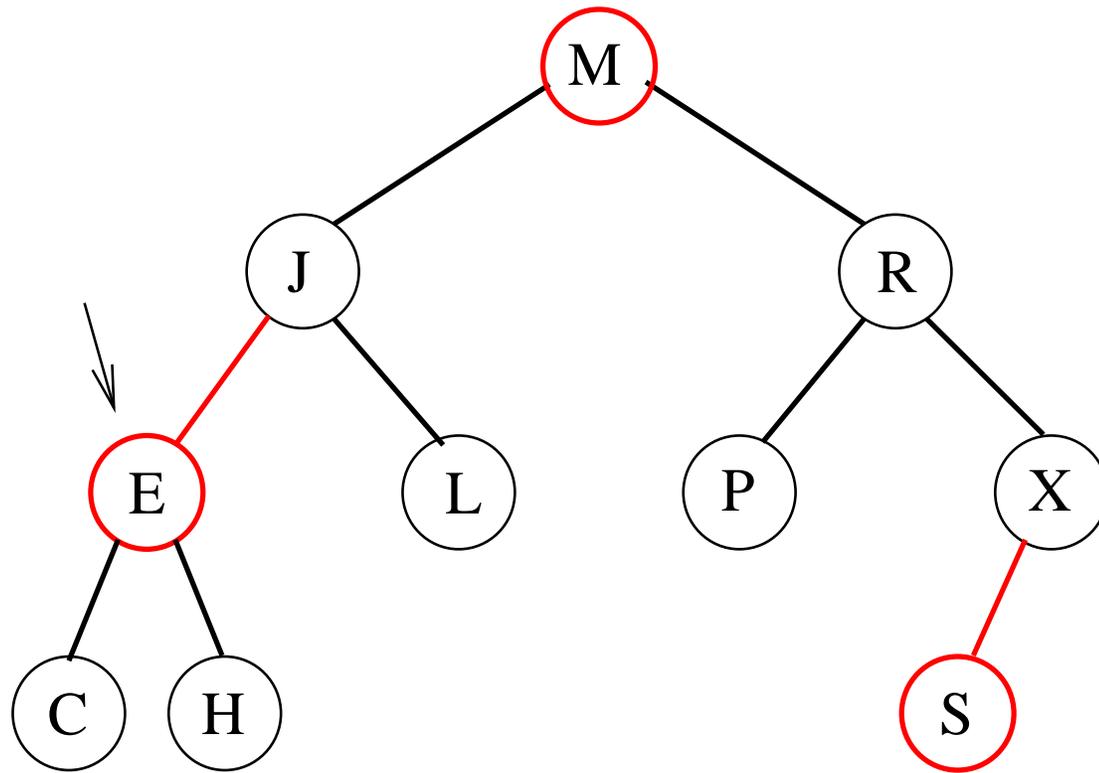
Exemplo

Remove C.



Exemplo

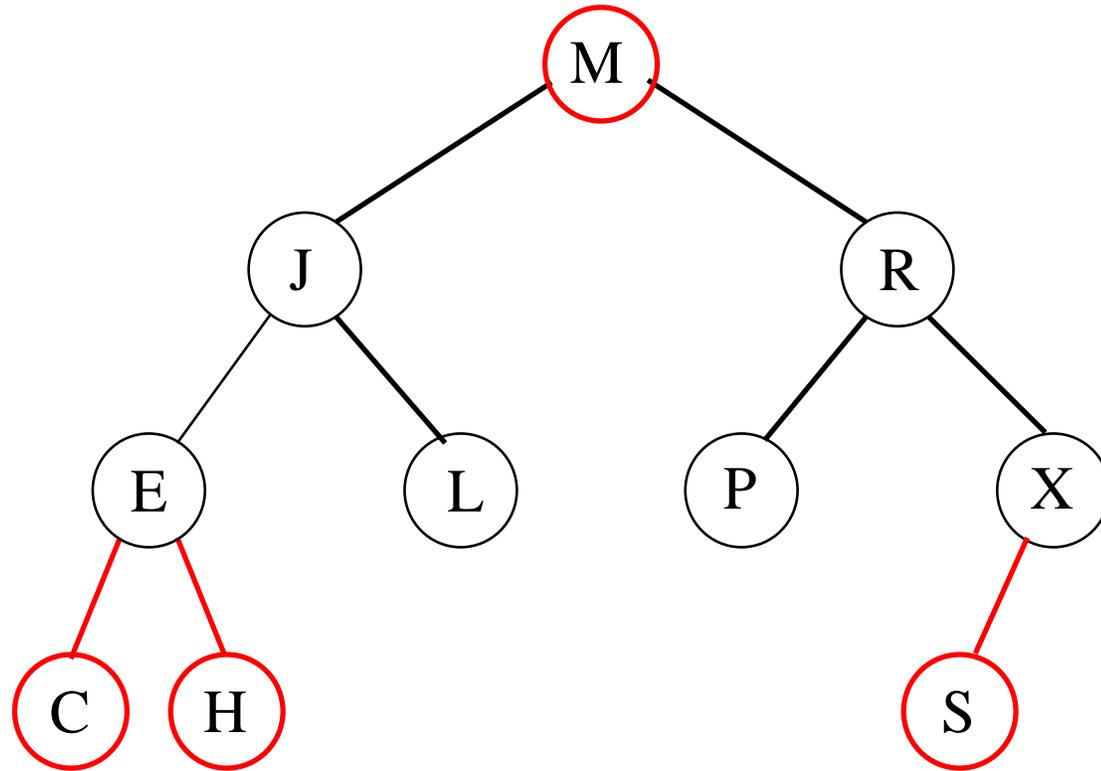
Remova C.



MOVARUBROESQ

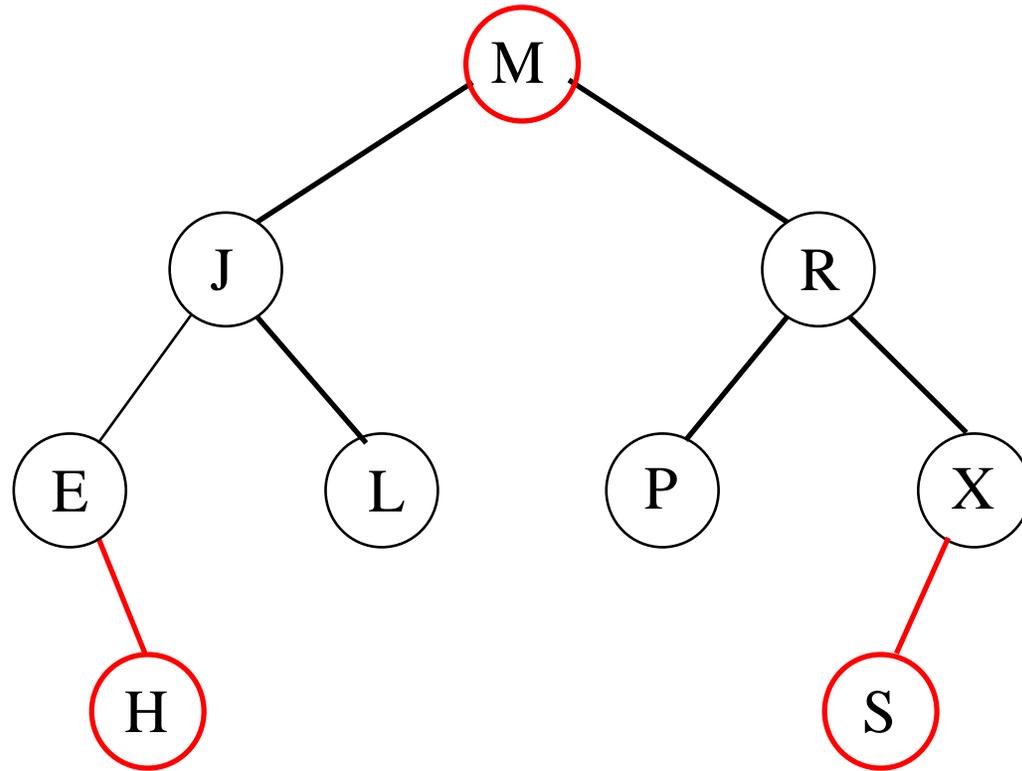
Exemplo

Remove C.



Exemplo

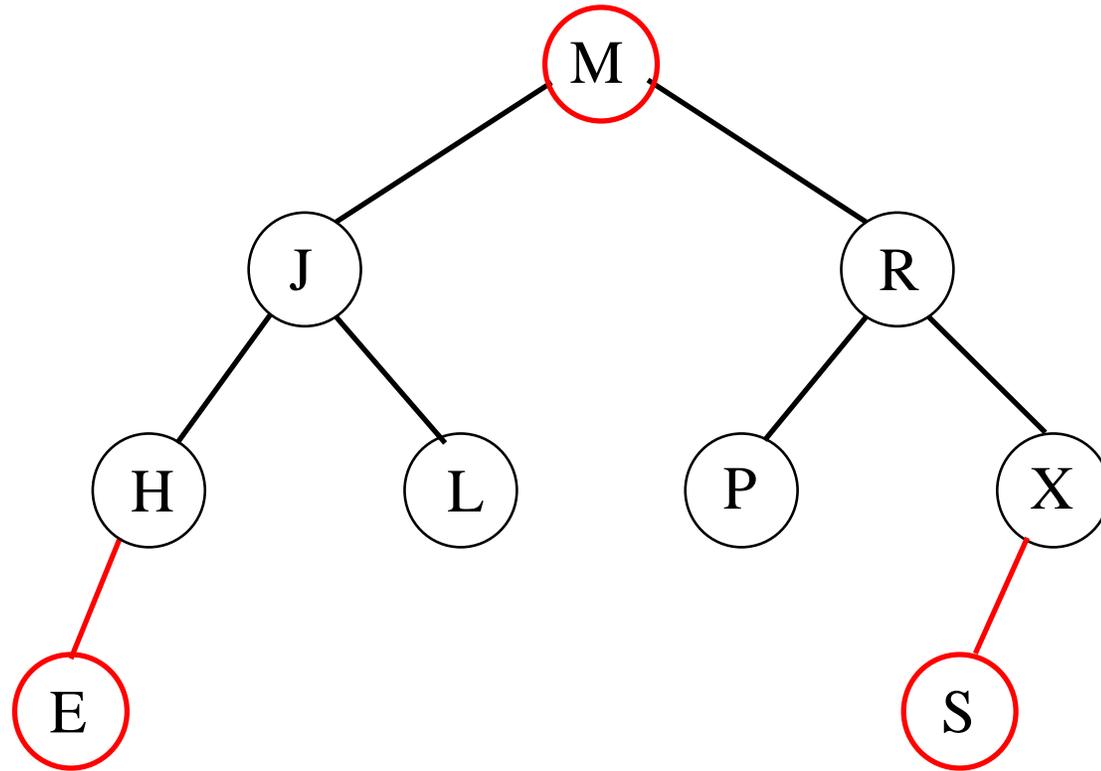
Remova C.



BALANCEIE

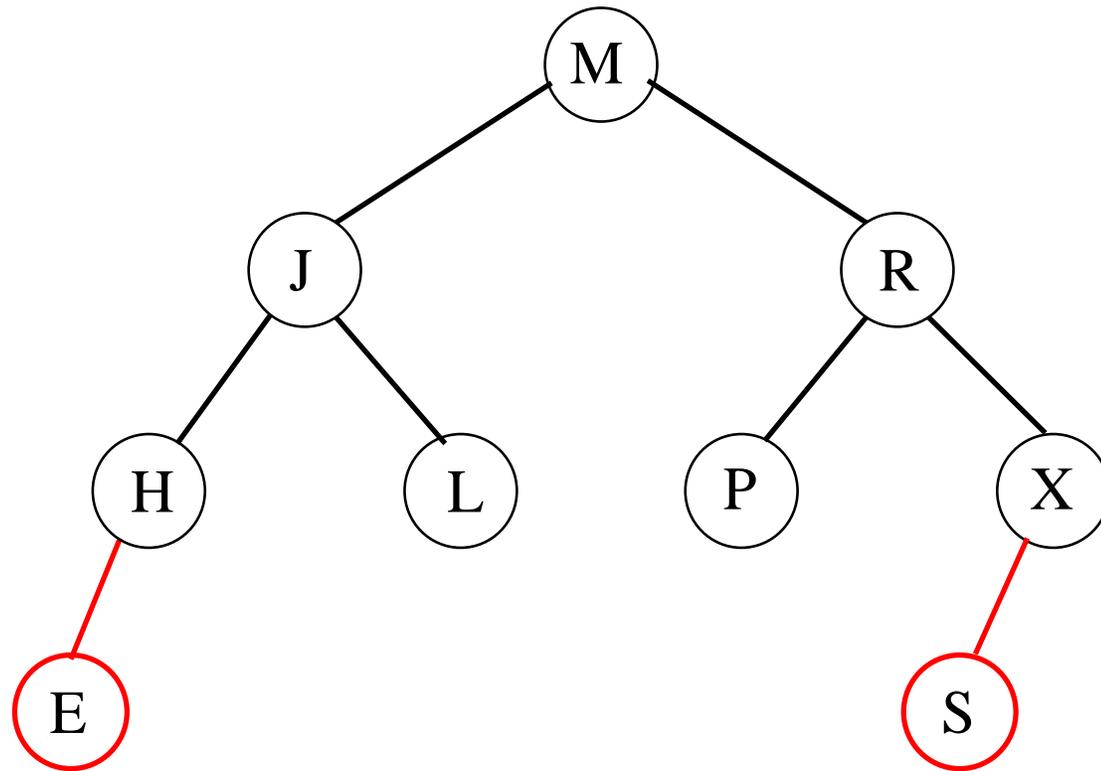
Exemplo

Remova C.



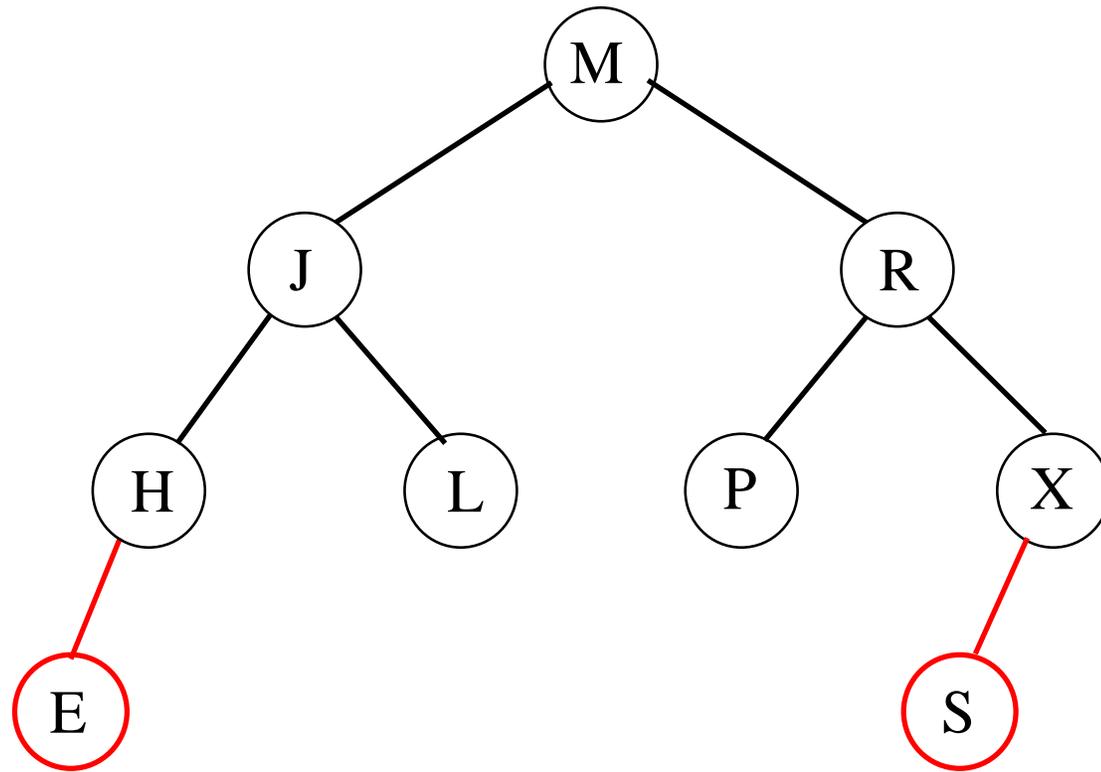
Exemplo

Remova C.



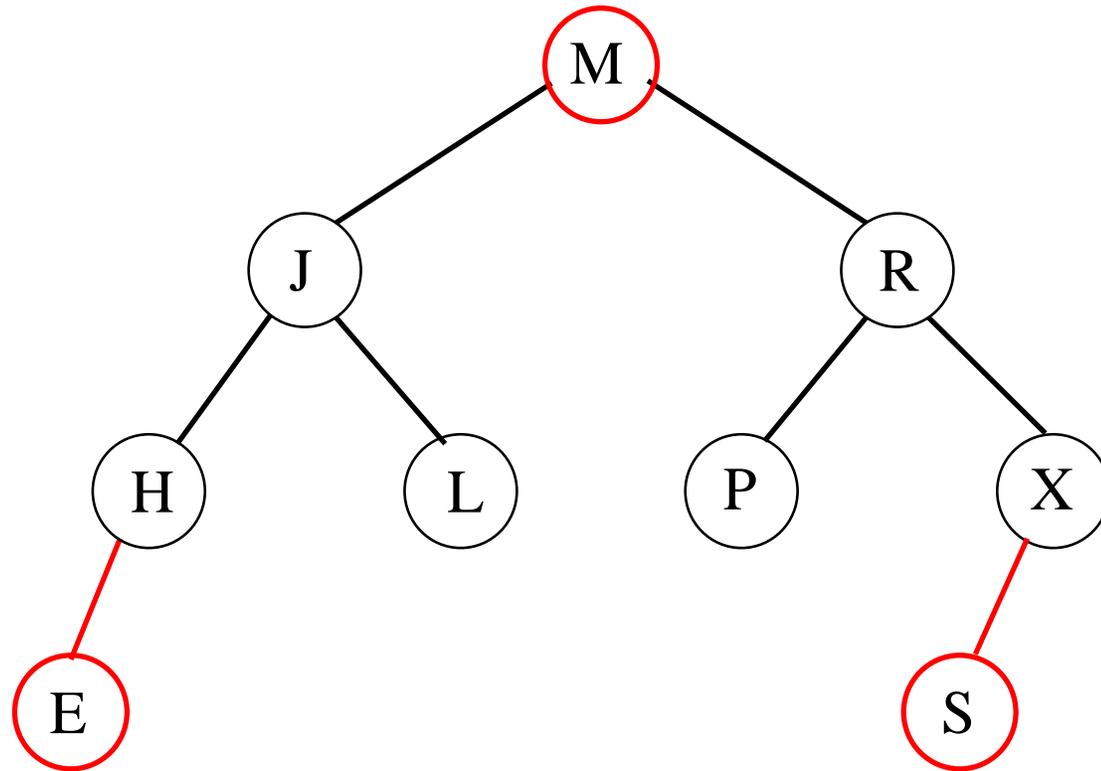
Exemplo

Remova E.



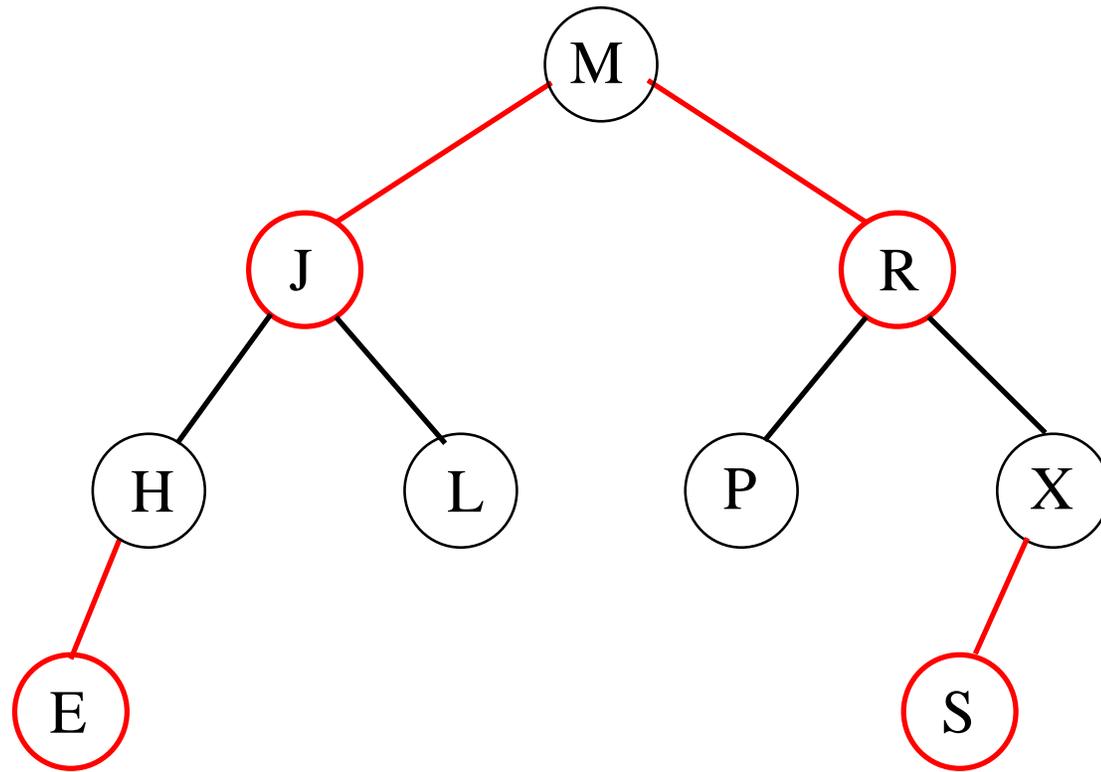
Exemplo

Remova E.



Exemplo

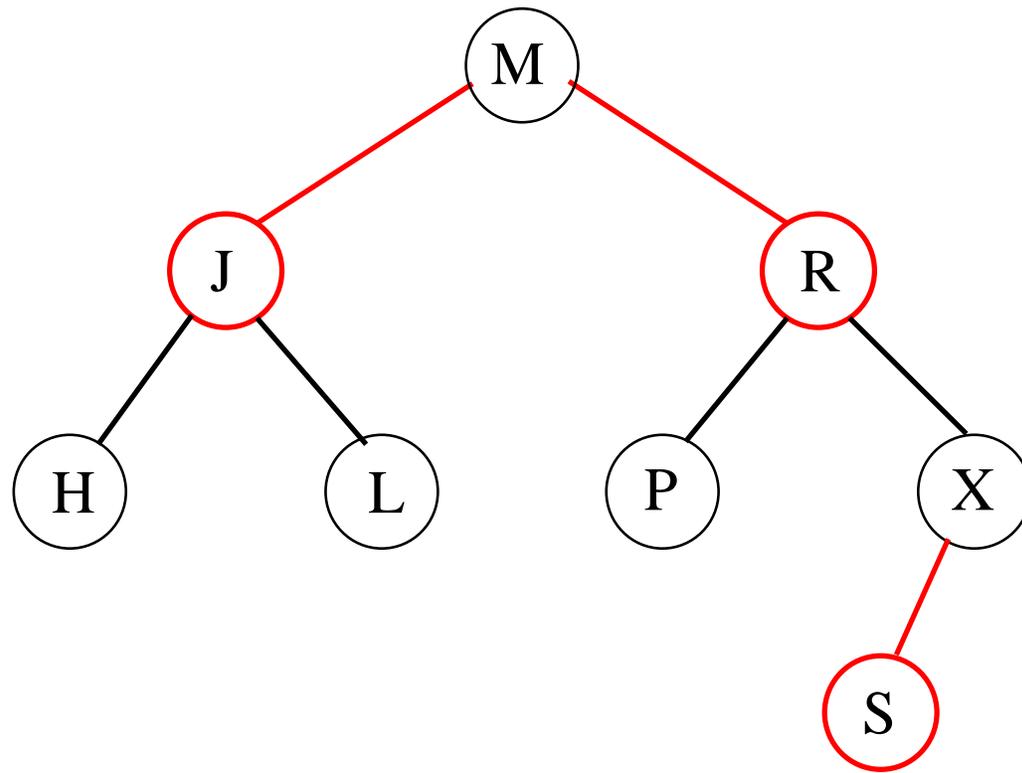
Remova E.



MOVARUBROESQ

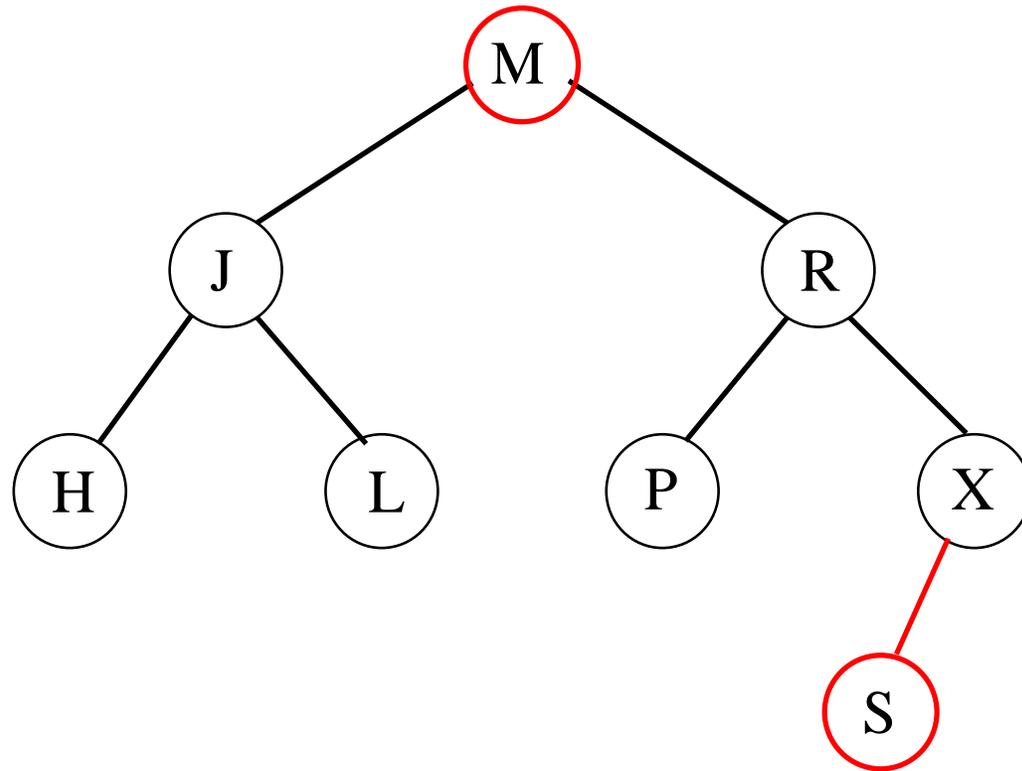
Exemplo

Remova E.



Exemplo

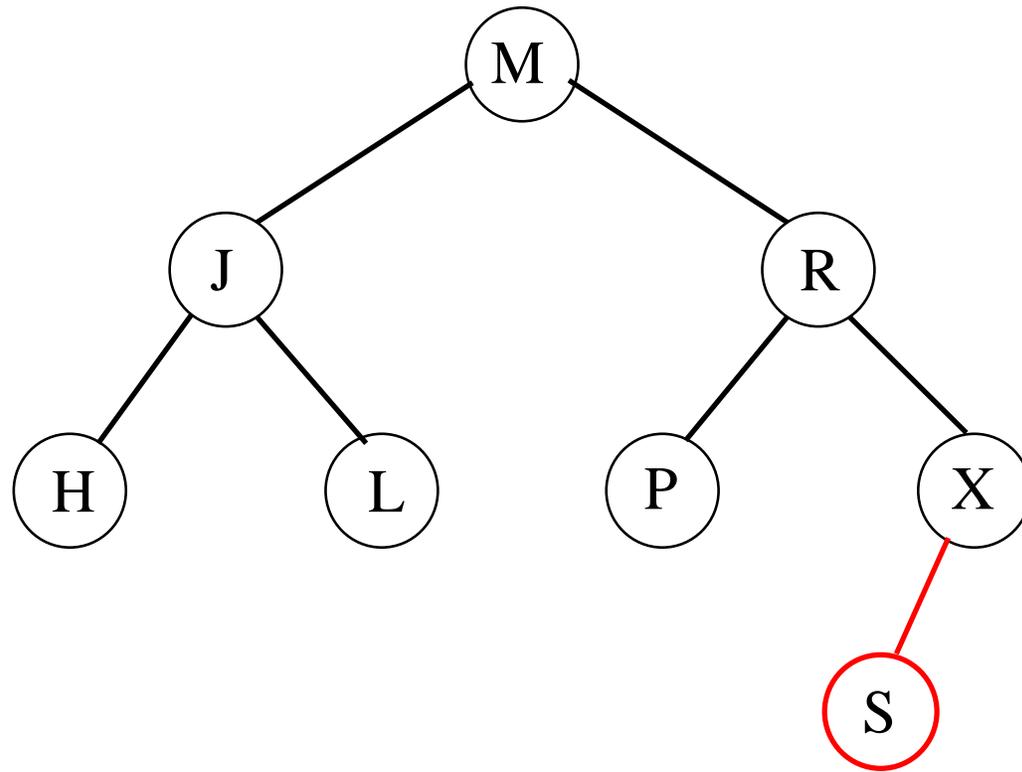
Remova E.



BALANCEIE

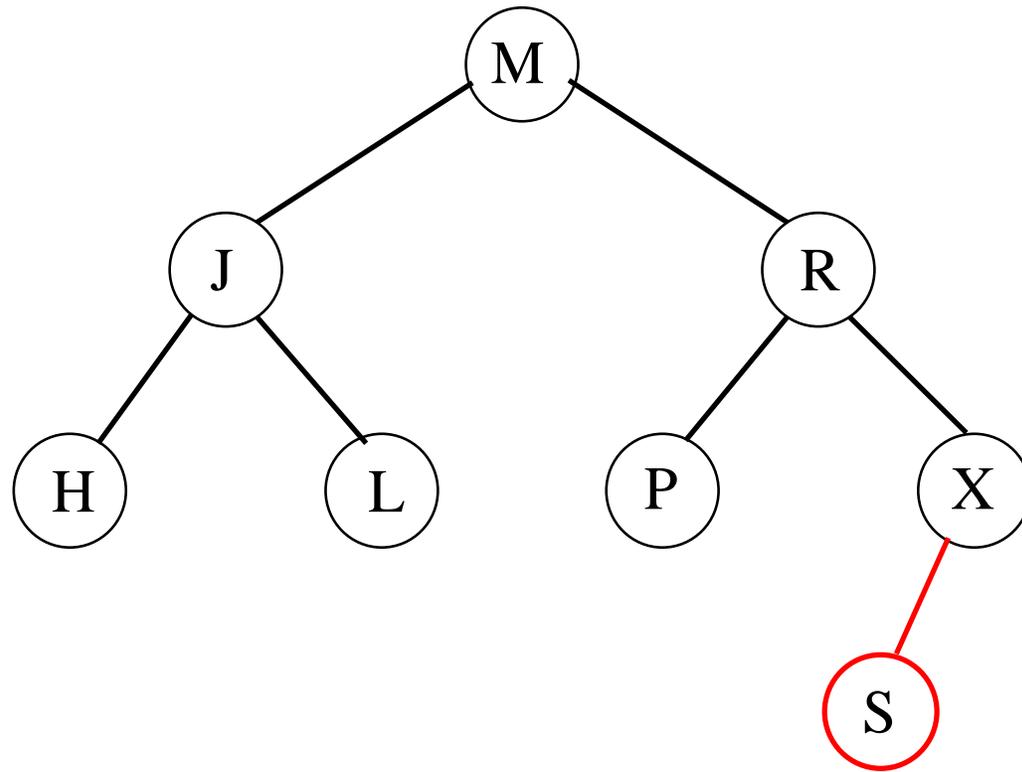
Exemplo

Remova E.



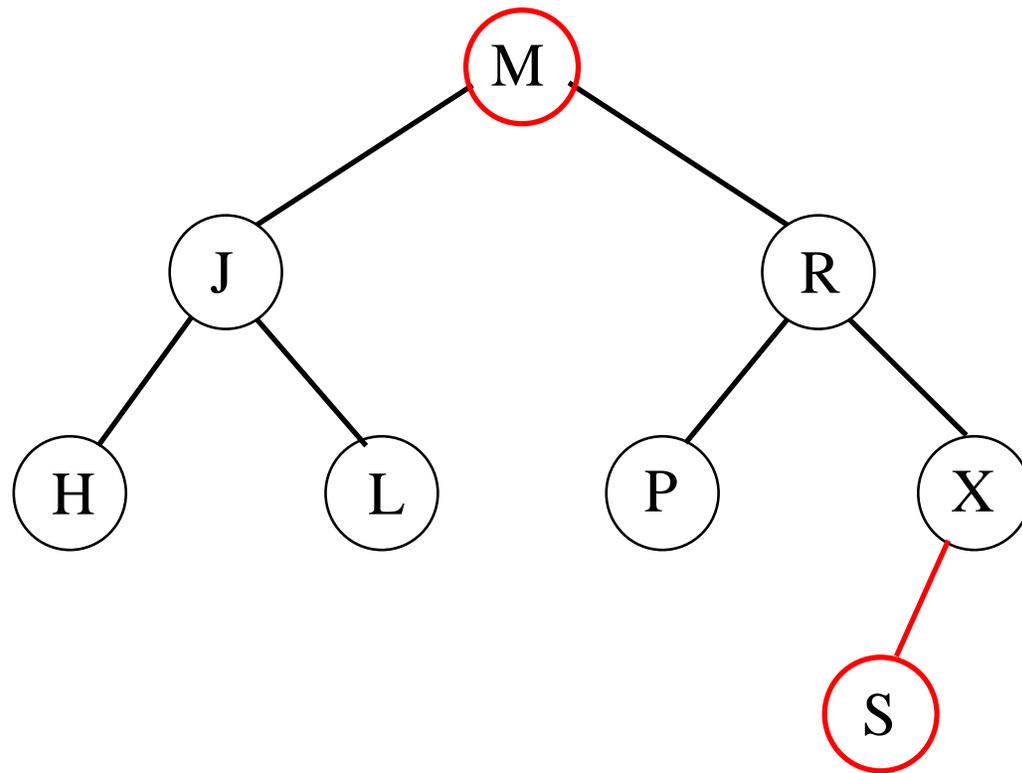
Exemplo

Remova H.



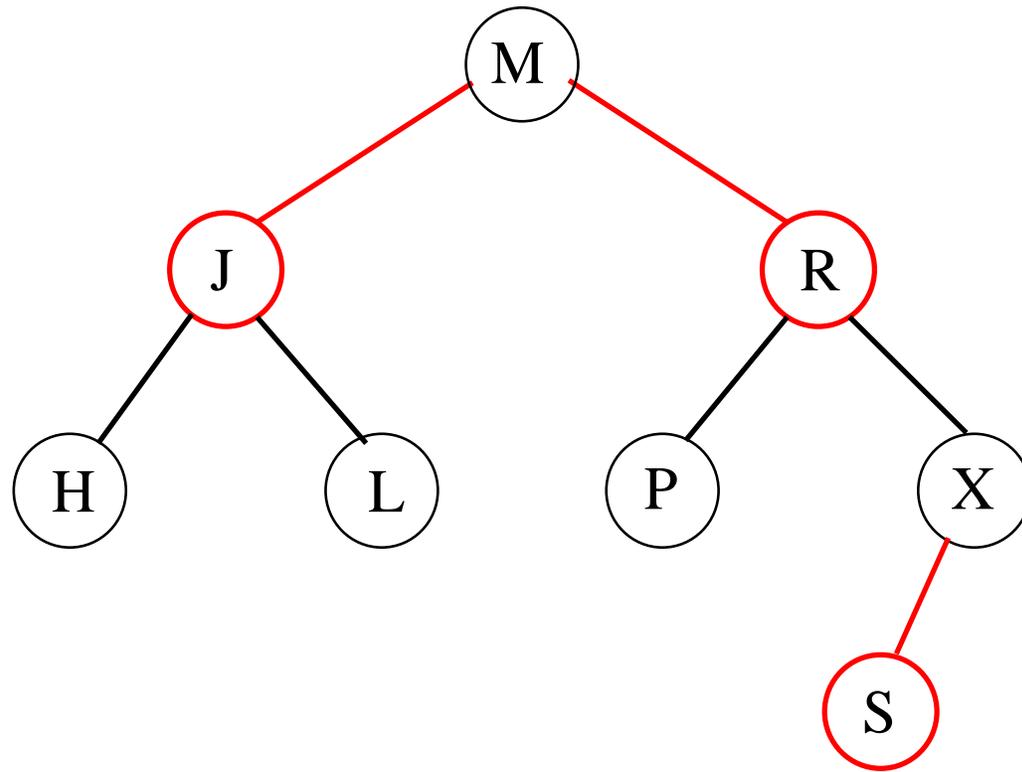
Exemplo

Remova H.



Exemplo

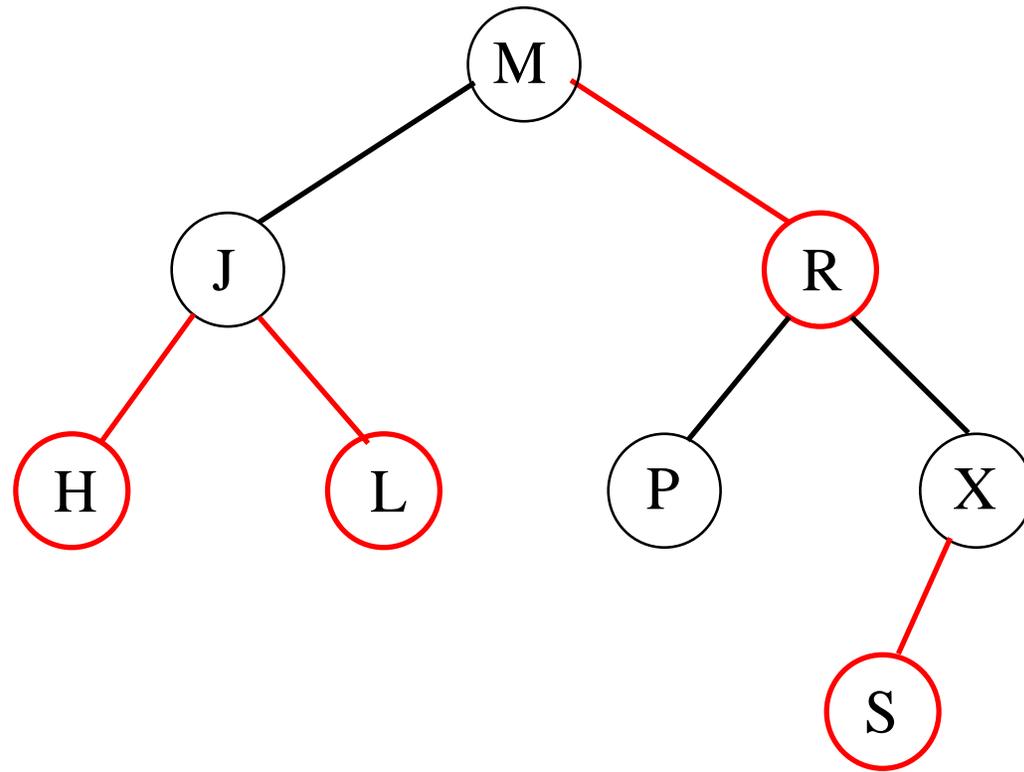
Remova H.



MOVARUBROESQ

Exemplo

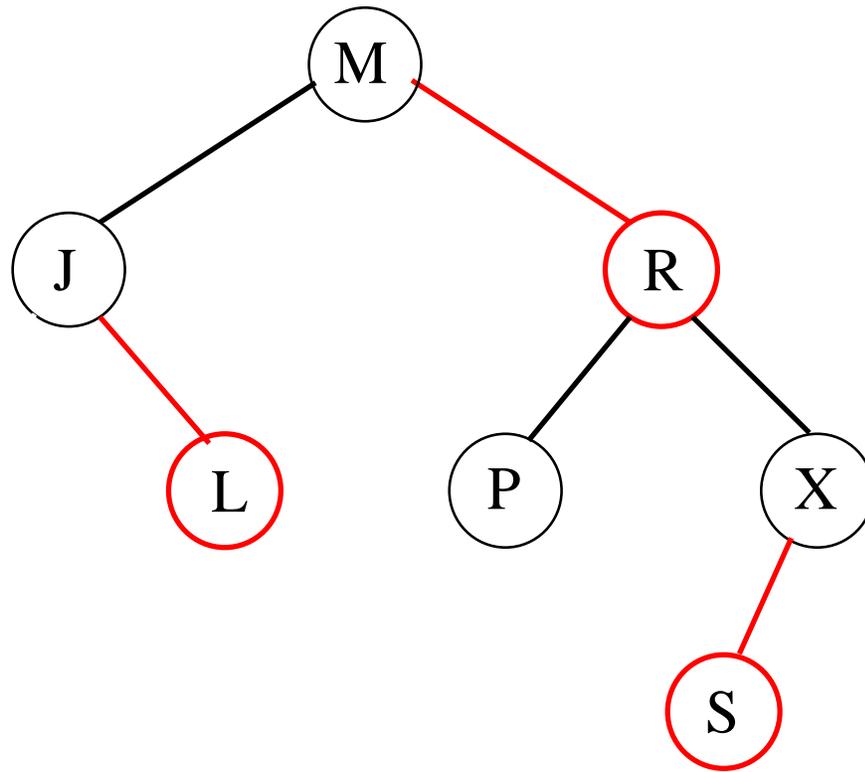
Remova H.



MOVARUBROESQ

Exemplo

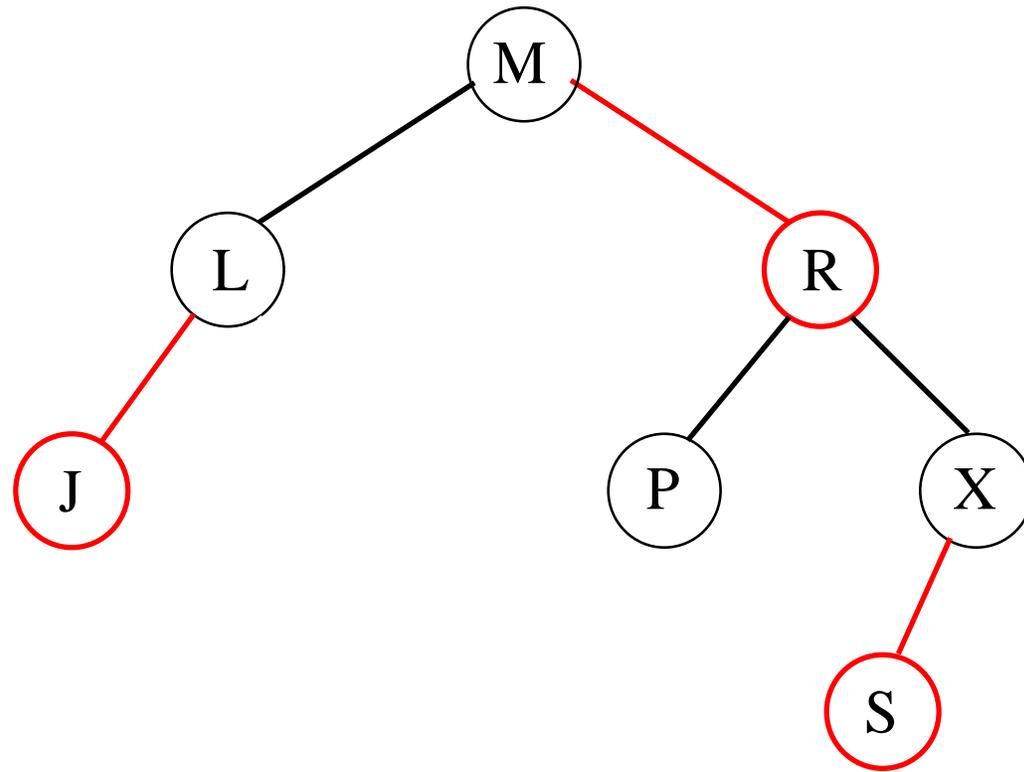
Remova H.



BALANCEIE

Exemplo

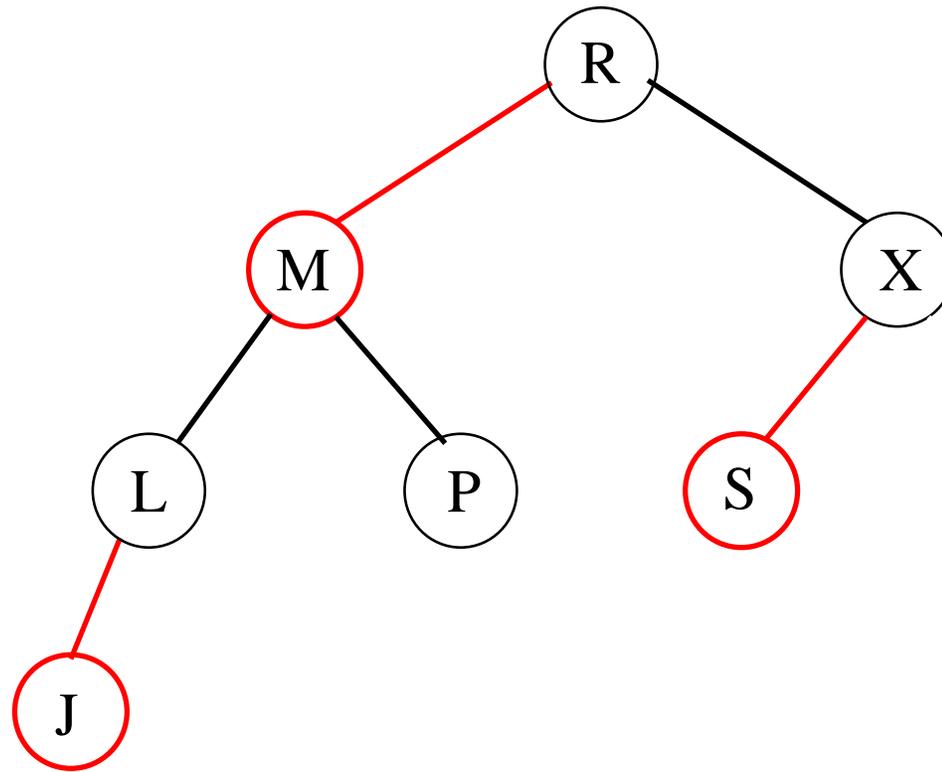
Remova H.



BALANCEIE

Exemplo

Remova H.



BALANCEIE

Comentários finais

Exercício: Simule a inserção em uma árvore rubro-negra inicialmente vazia das seguintes chaves:

20, 17, 38, 40, 53, 10, 6, 16, 23, 14, 11, 50, 45.

Simule a remoção de alguns destes valores.

Quarta edição do livro do Sedgwick (com Wayne):

<http://www.cs.princeton.edu/algs4/>

Veja em particular algumas animações neste site.