

ÁRVORES BINÁRIAS

Balanceamento

Prof: Bruna Moreira
brunamoreira@unb.br



ÁRVORES BINÁRIAS DE BUSCA

Não-linear

Estrutura composta por nós

Menores à esquerda

Maiores à direita

Ponteiro para filhos (2)

ROOT - primeiro nó

LEAVES - últimos nós

armazenamento
não contíguo

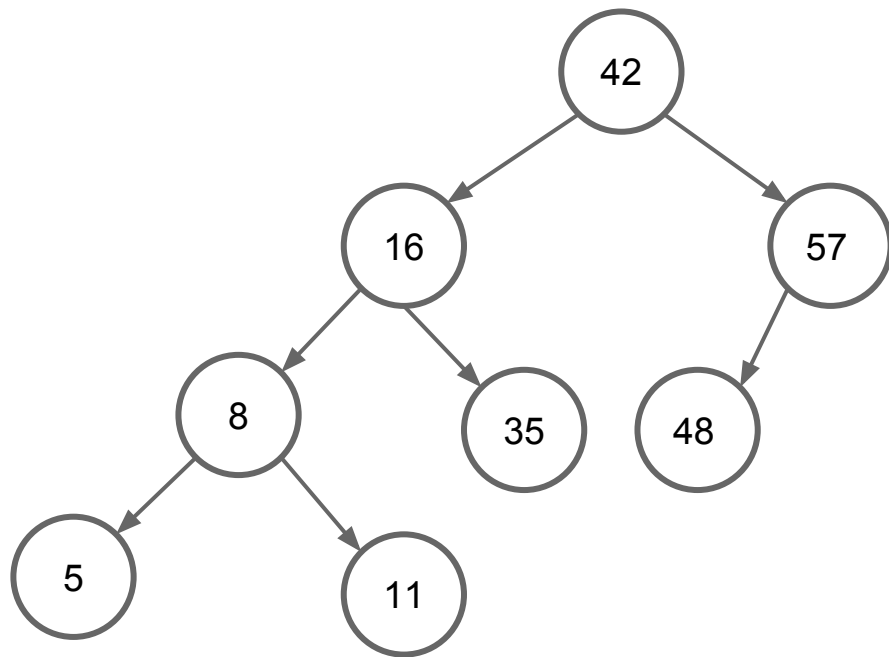
facilita
ordenação

Árvore binária

diminuir
complexidade
de busca

conceito

Árvore binária

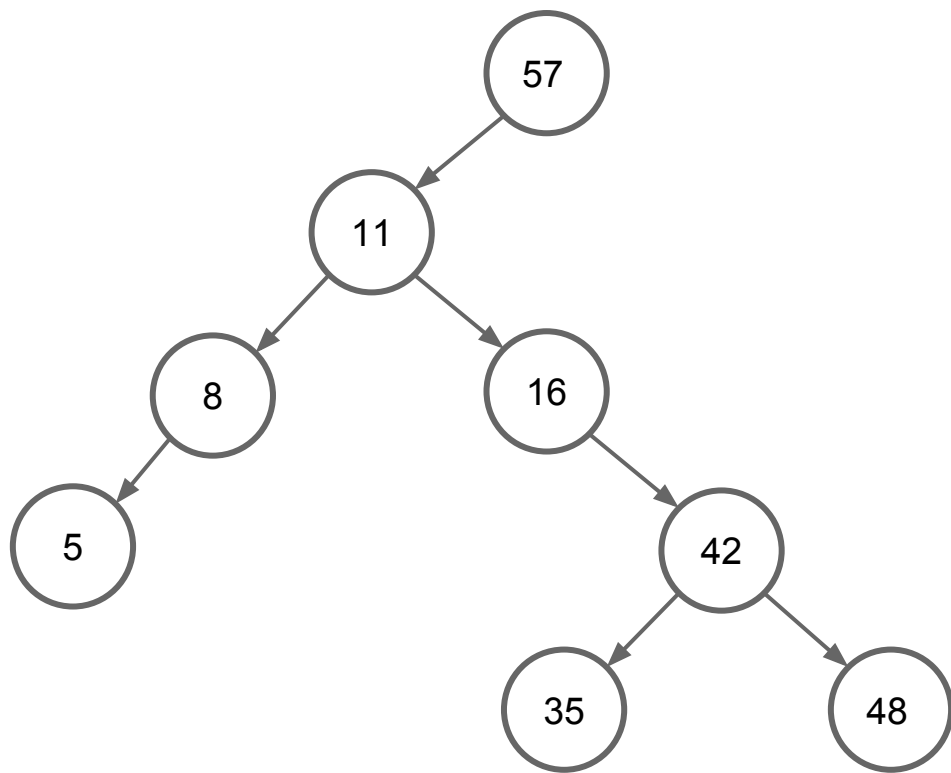


Raiz

Menores <-

Maiores ->

0 a 2 filhos



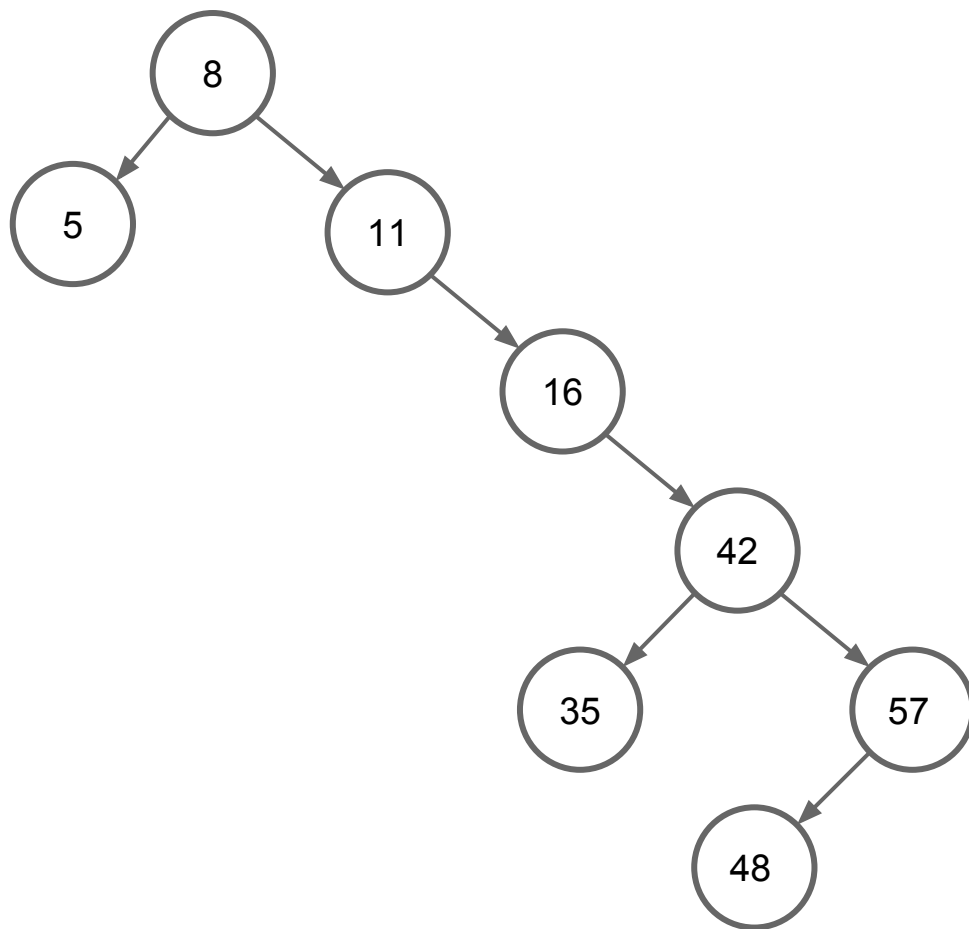
Árvore binária

Raiz

Menores <-

Maiores ->

0 a 2 filhos



Árvore binária

Raiz

Menores <-

Maiores ->

0 a 2 filhos

O DESBALANCEAMENTO PREJUDICA A BUSCA

Pode virar $O(n)$

O DESBALANCEAMENTO PRETUDICA BUSCA

Pode virar

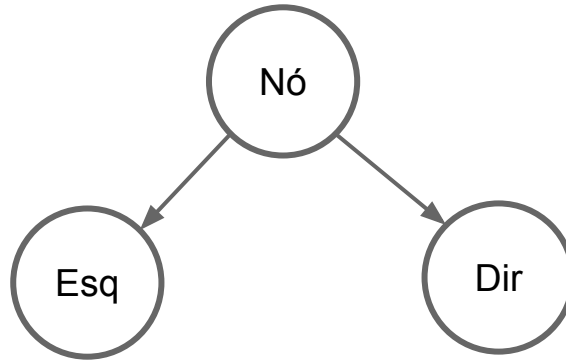
Quando a árvore
está
desbalanceada?

ÁRVORE BALANCEADA

diferença de altura
das duas subárvores
de qualquer nó é
menor ou igual a 1

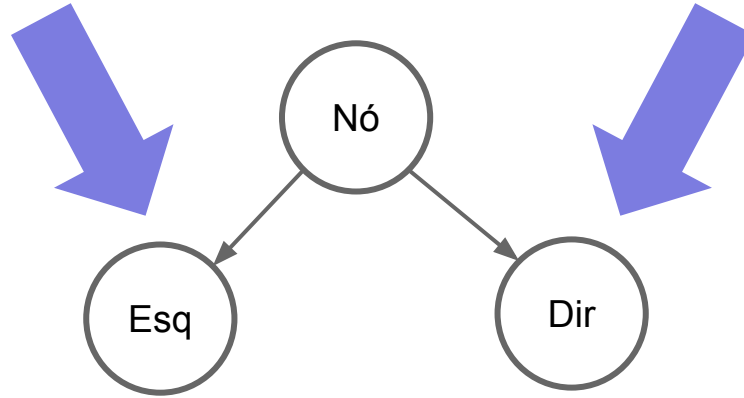
ÁRVORE BALANCEADA

diferença de altura
das duas subárvores
de qualquer nó é
menor ou igual a 1



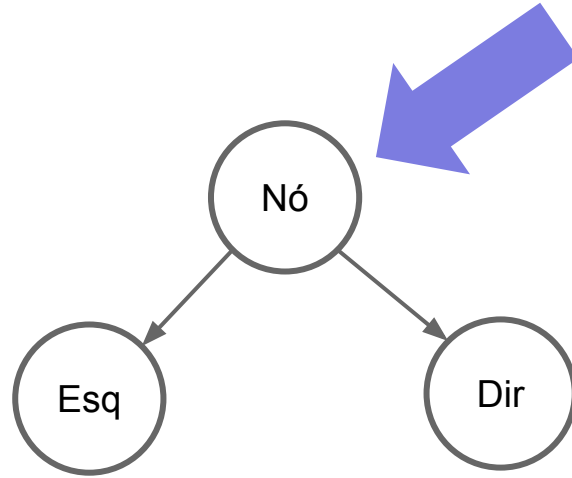
ÁRVORE BALANCEADA

diferença de altura
das duas **subárvores**
de qualquer nó é
menor ou igual a 1



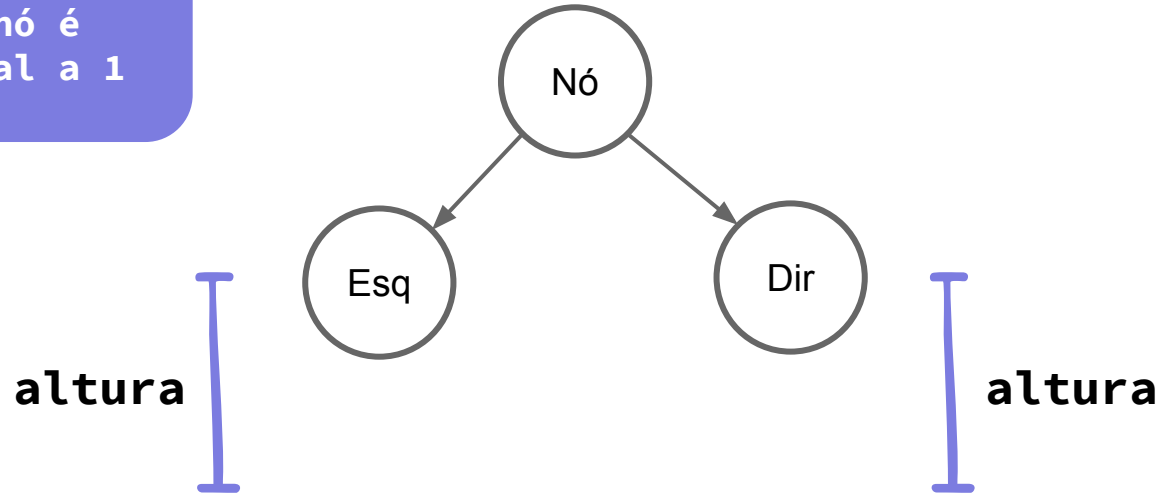
ÁRVORE BALANCEADA

diferença de altura
das duas subárvores
de qualquer **nó** é
menor ou igual a 1



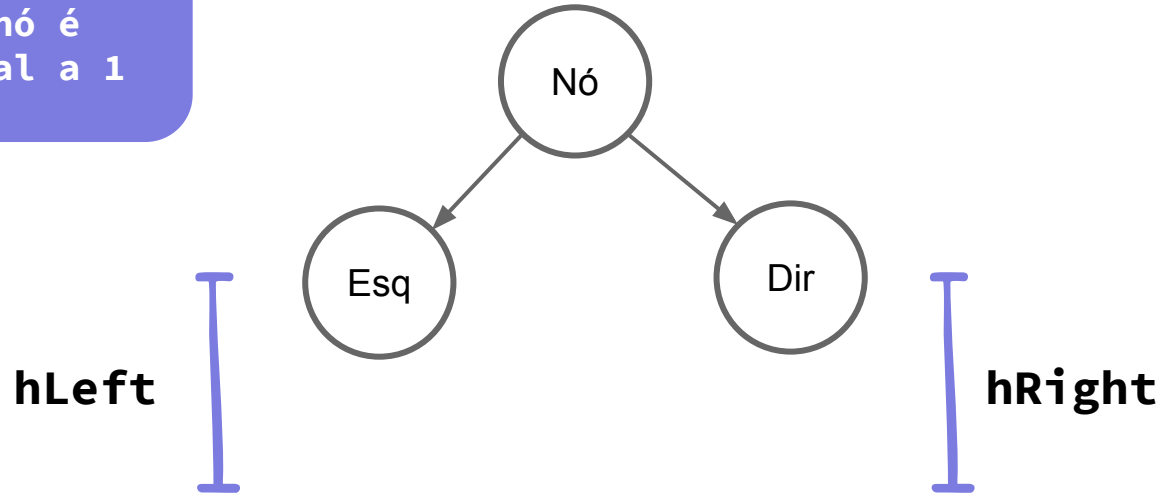
ÁRVORE BALANCEADA

diferença de **altura**
das duas subárvores
de qualquer nó é
menor ou igual a 1



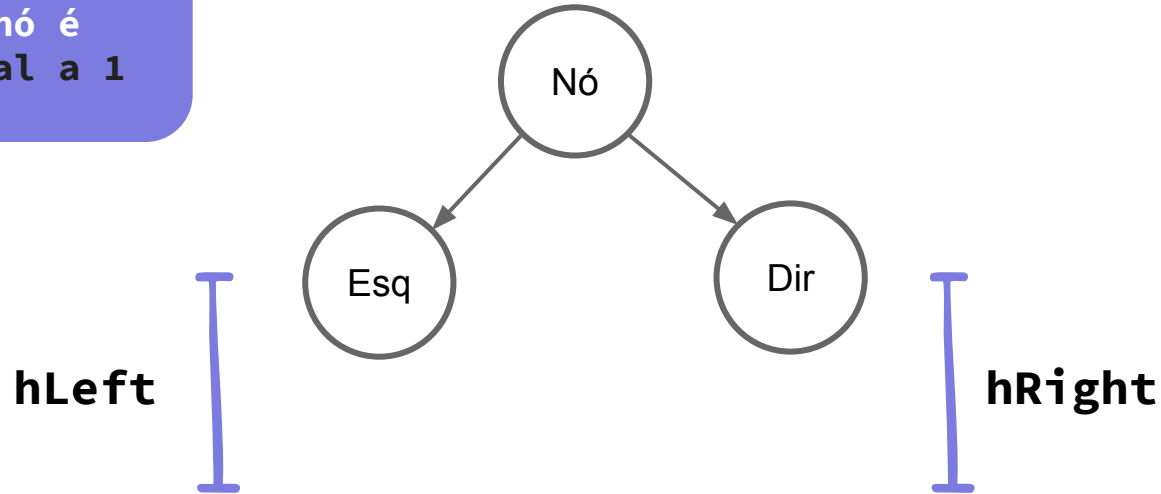
ÁRVORE BALANCEADA

diferença de **altura**
das duas subárvores
de qualquer nó é
menor ou igual a 1



ÁRVORE BALANCEADA

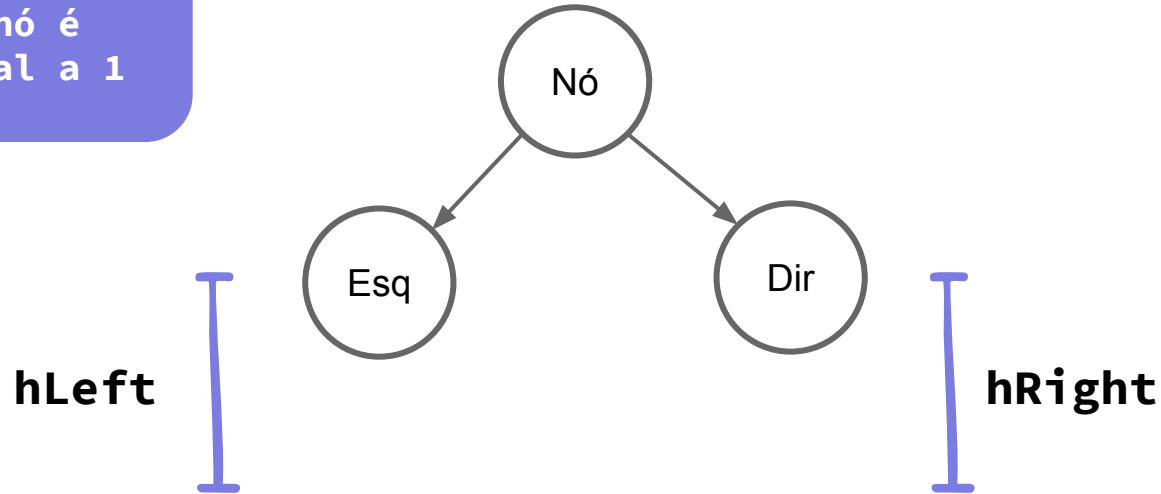
diferença de altura
das duas subárvores
de qualquer nó é
menor ou igual a 1



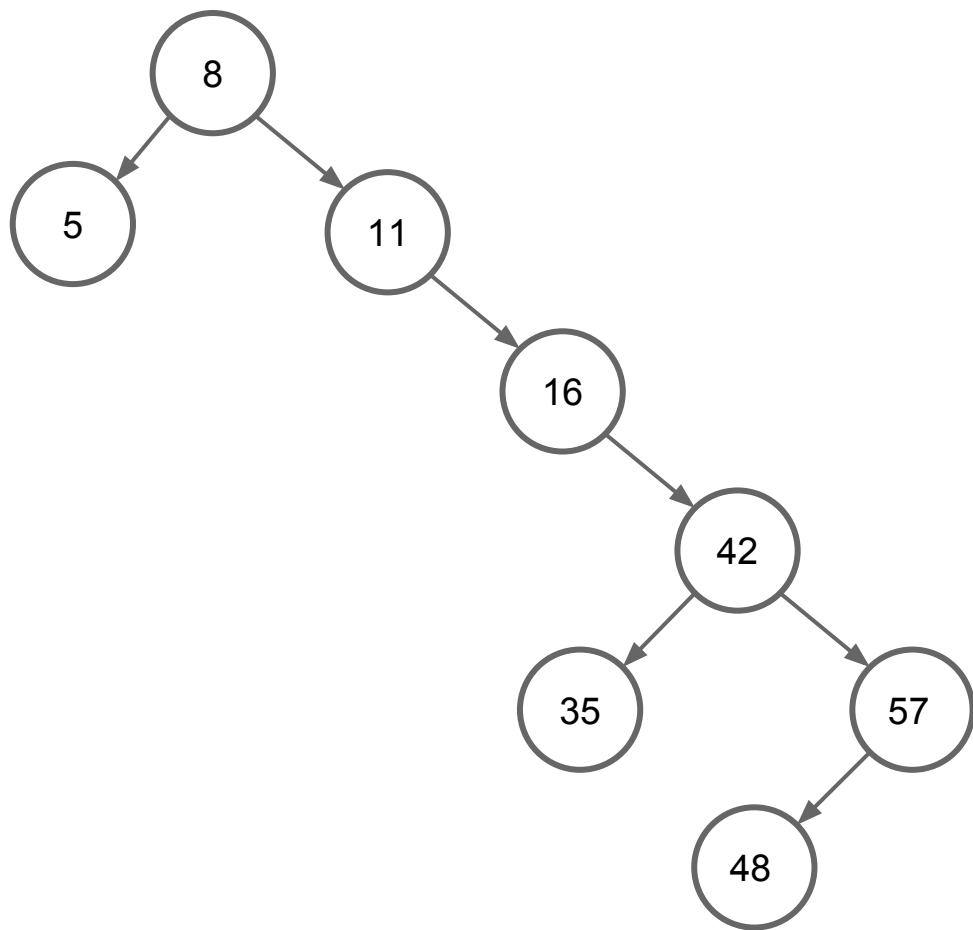
$$\text{abs}(h\text{Left} - h\text{Right}) \leq 1$$

ÁRVORE BALANCEADA

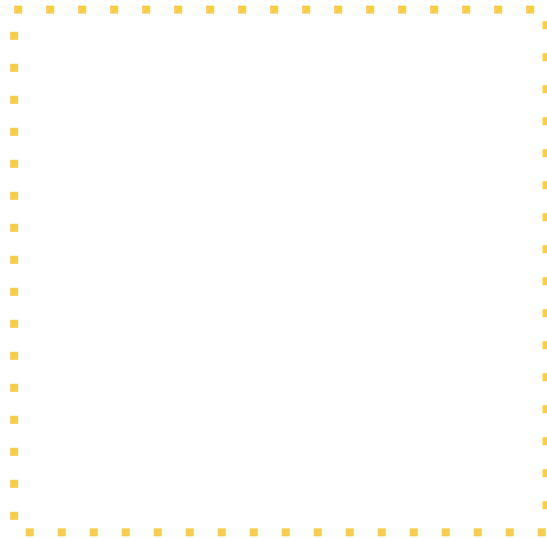
diferença de altura
das duas subárvores
de qualquer nó é
menor ou igual a 1

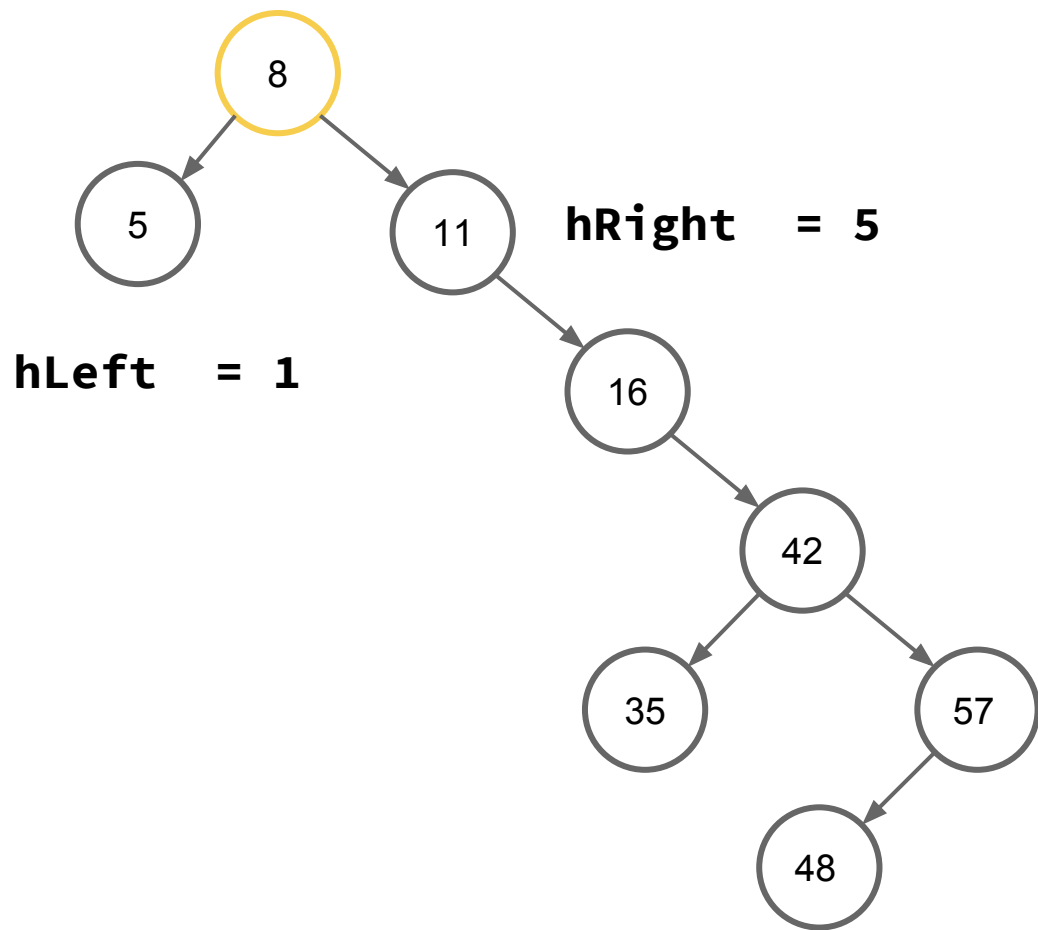


$$\text{abs}(h\text{Left} - h\text{Right}) \leq 1$$

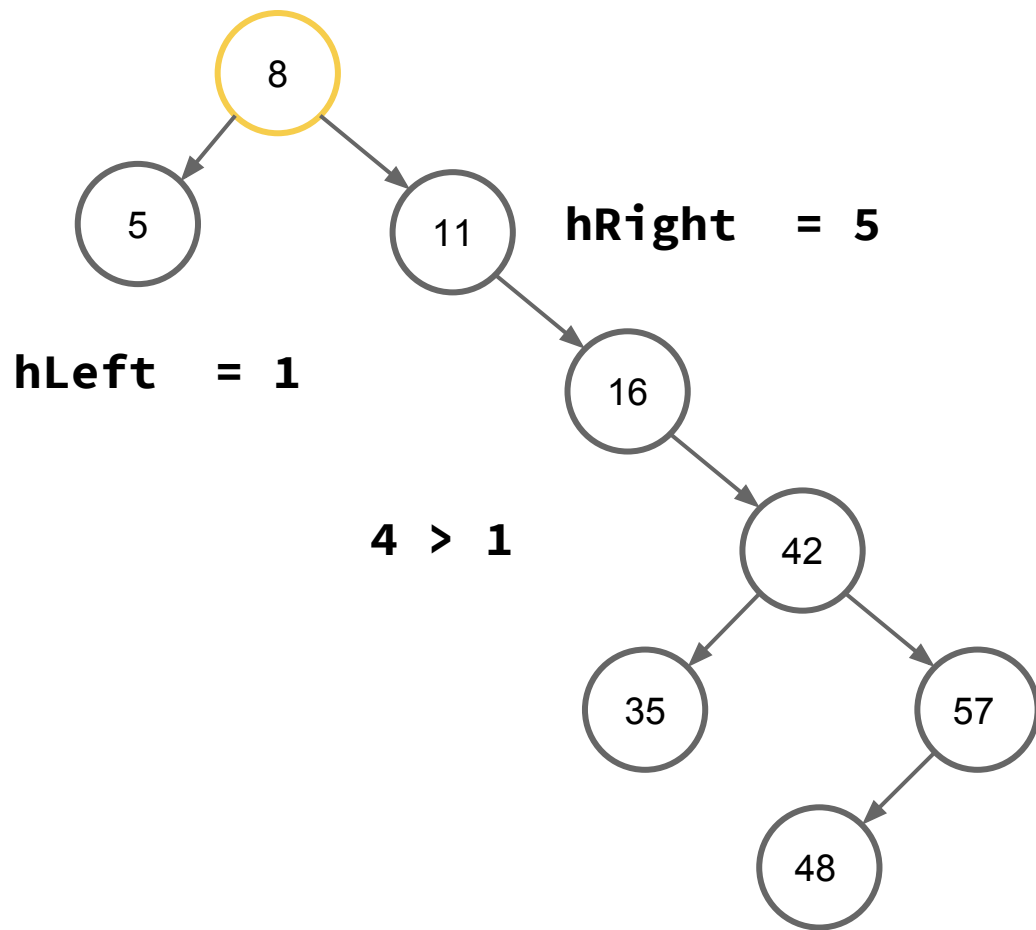


Balanceada?

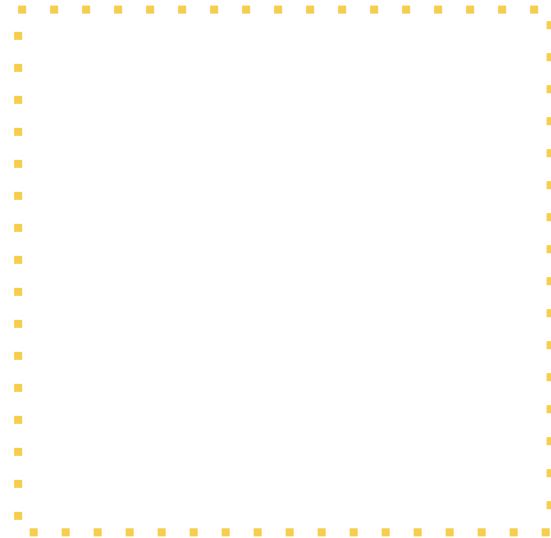


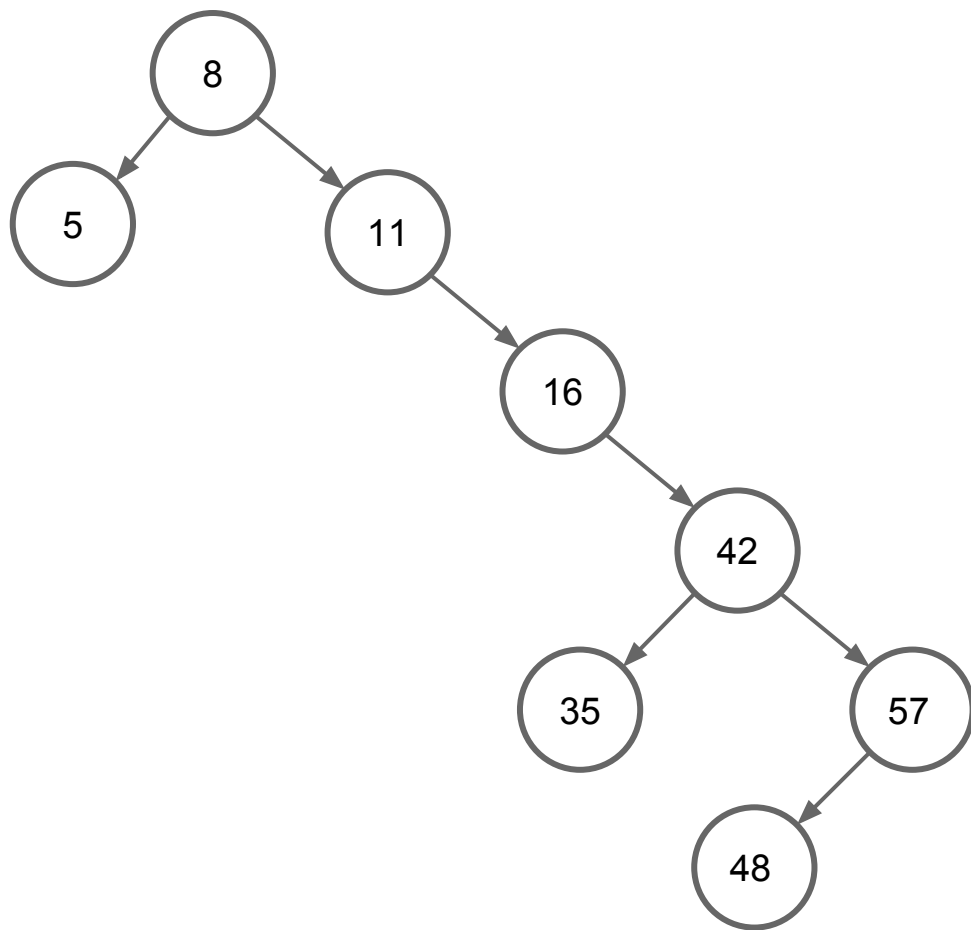


Balanceada?



Balanceada?



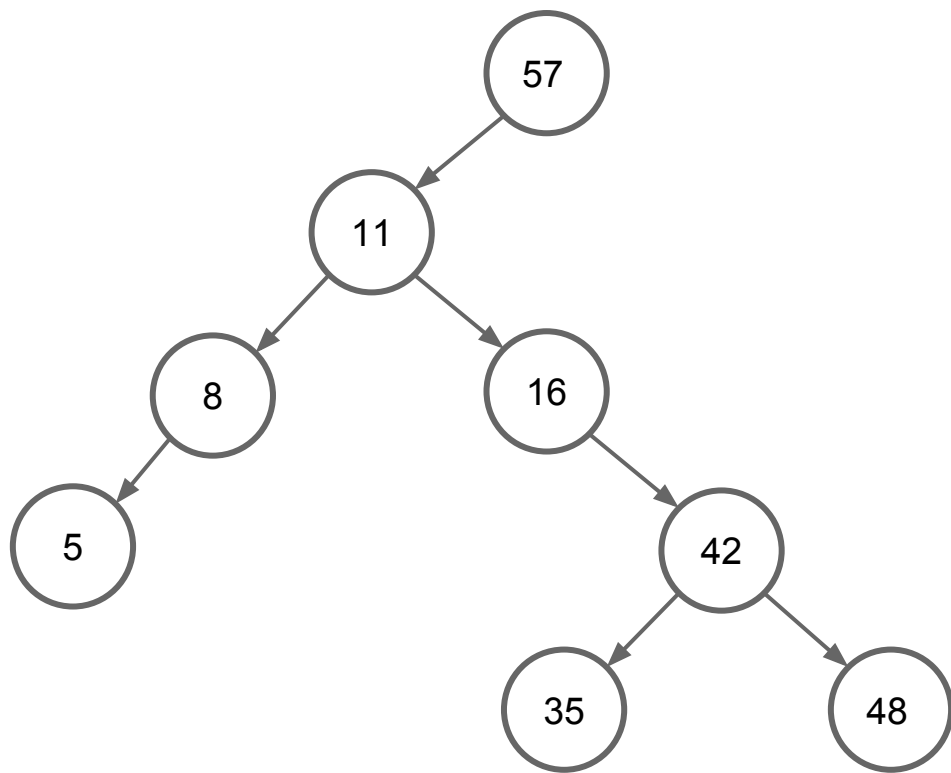


Desbalanceada

hRight = 5

hLeft = 1

4 > 1

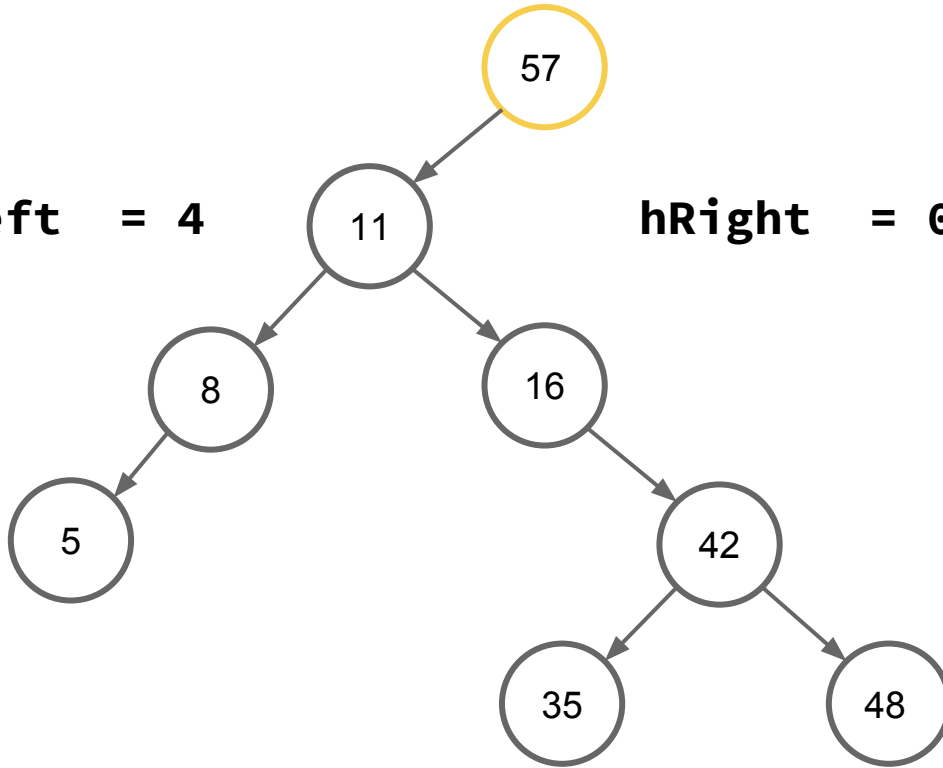


Balanceada?

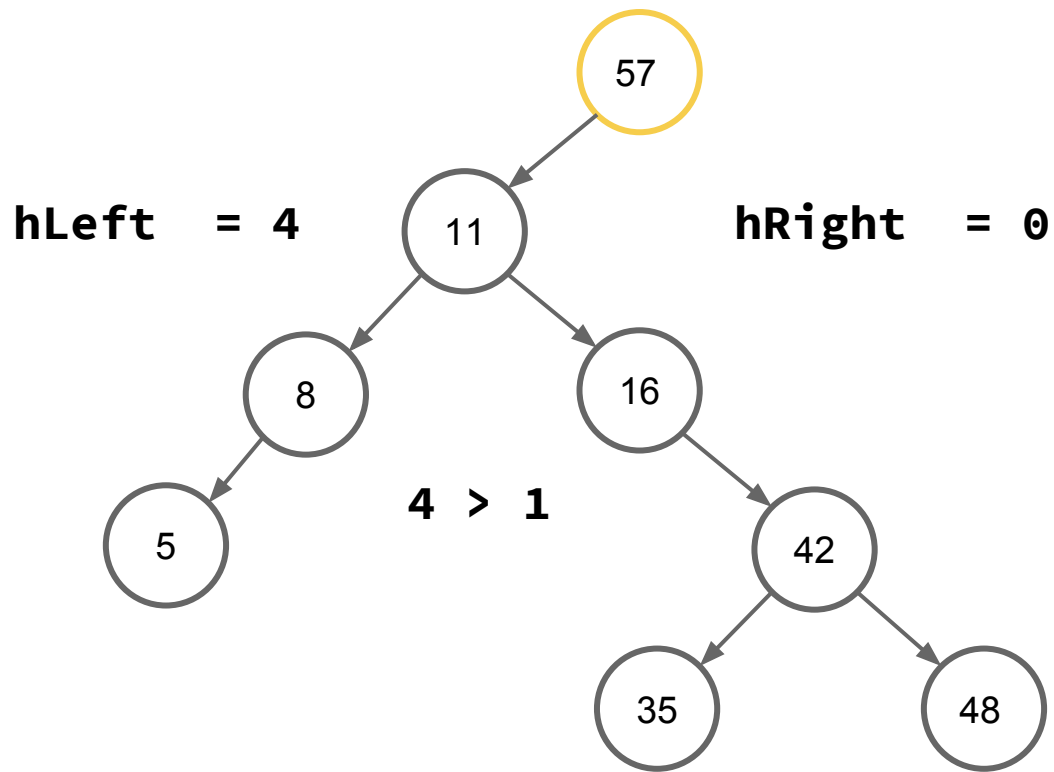


hLeft = 4

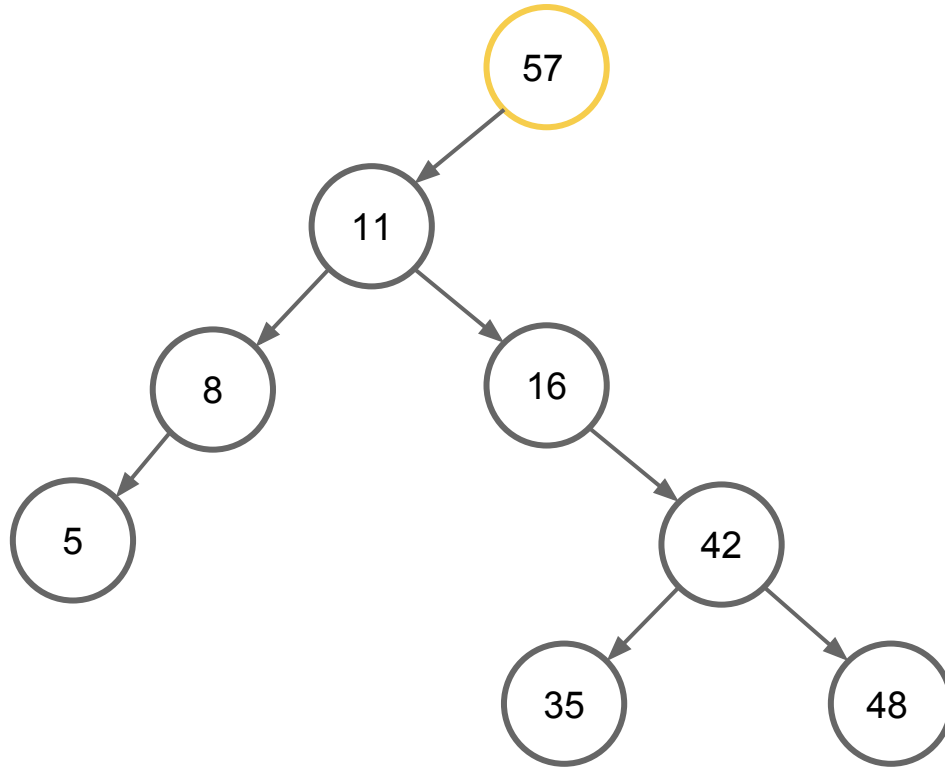
hRight = 0



Balanceada?



Balanceada?

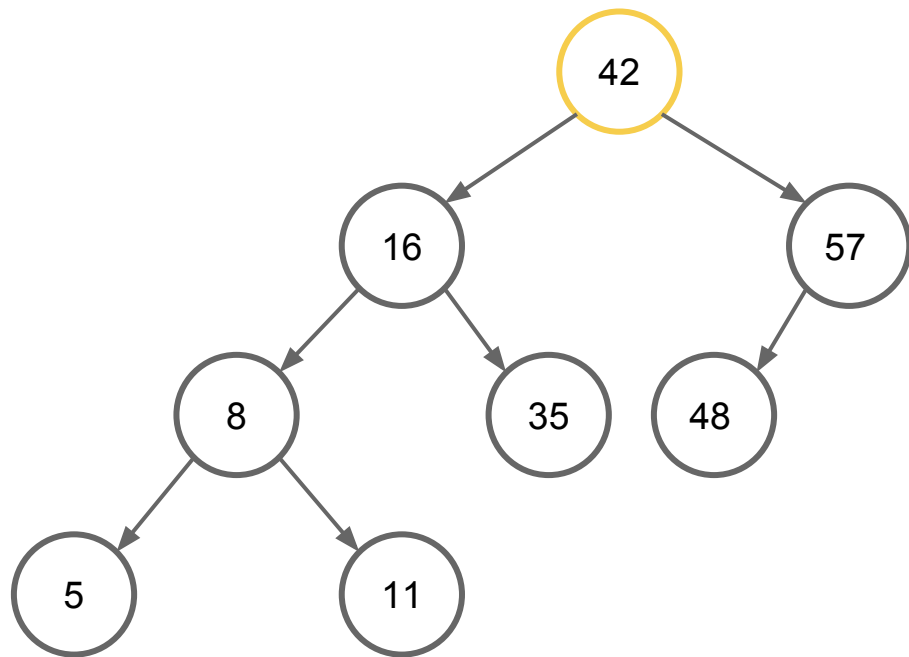


Desbalanceada

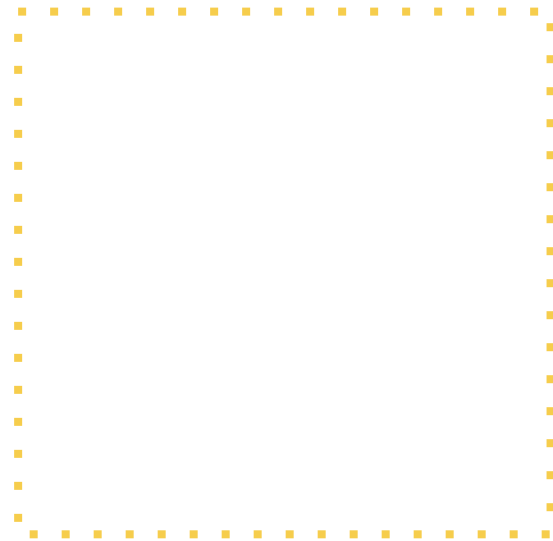
hRight = 4

hLeft = 0

4 > 1



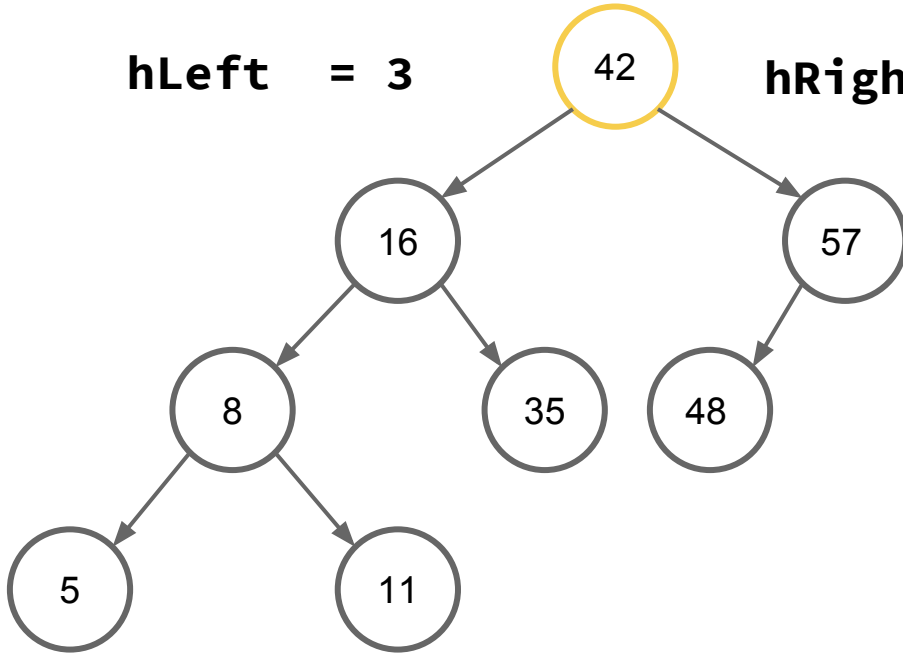
Balanceada?



hLeft = 3

42

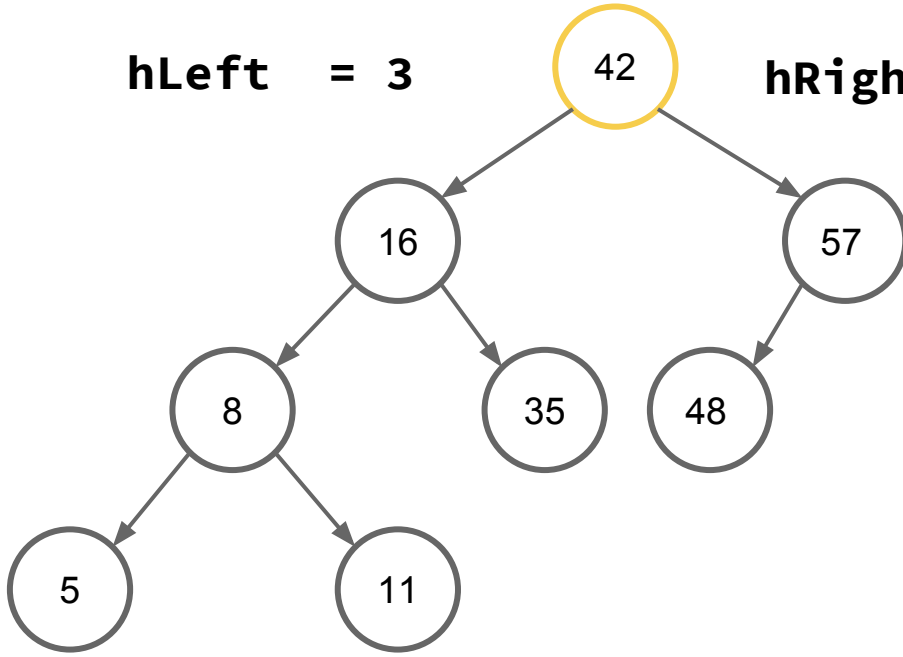
hRight = 2



Balanceada?

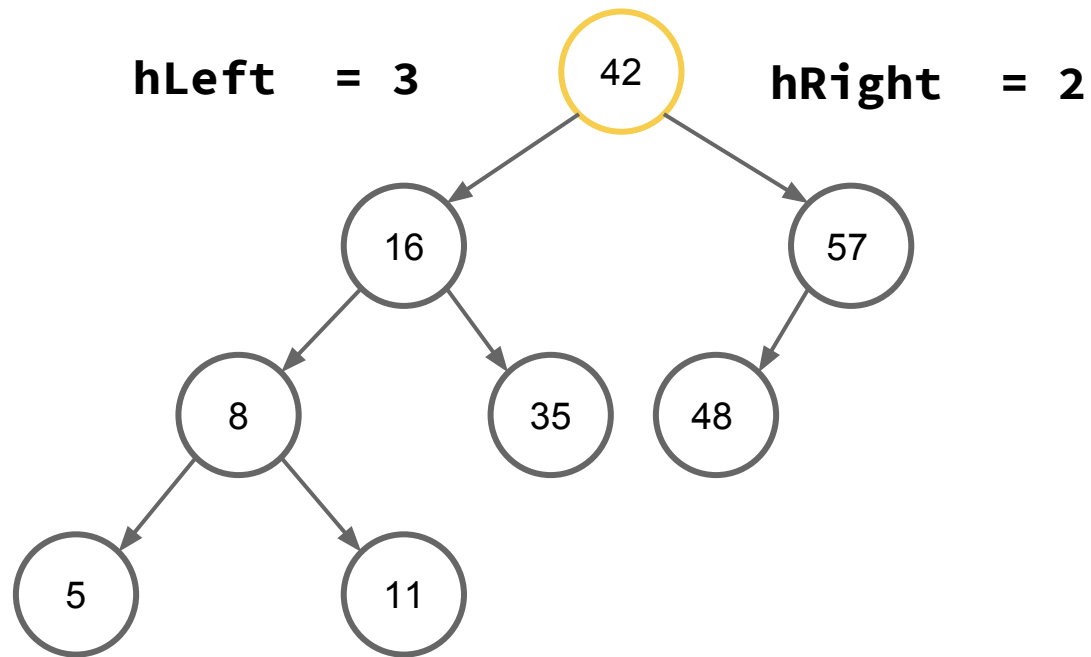
hLeft = 3

hRight = 2



Balanceada?

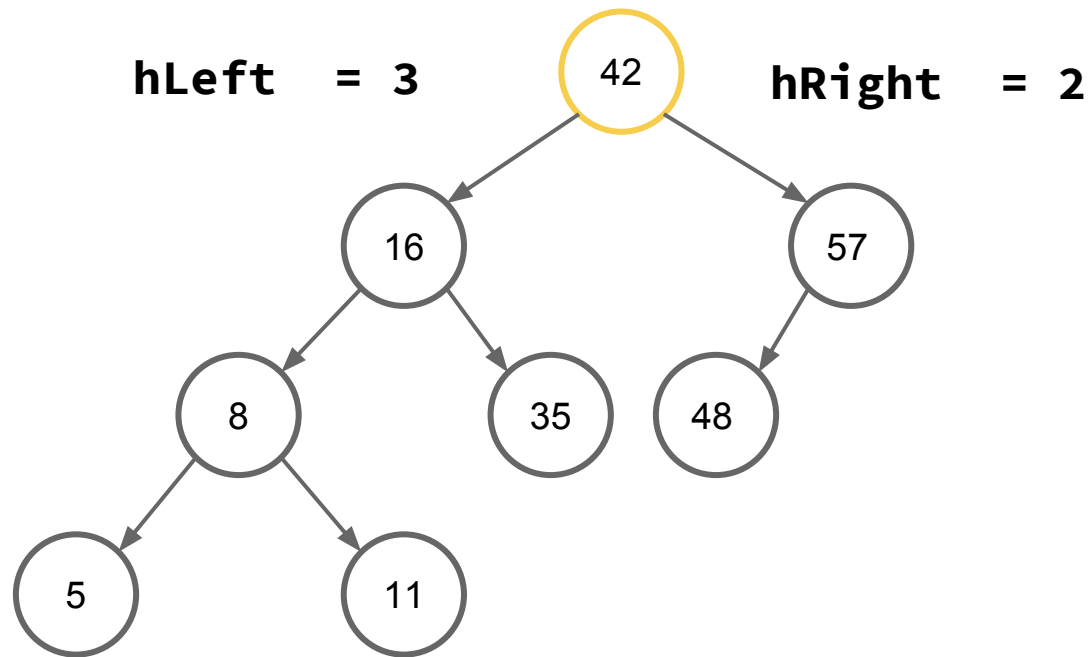
1 = 1



Balanceada?

diferença

1 = 1



Balanceada?

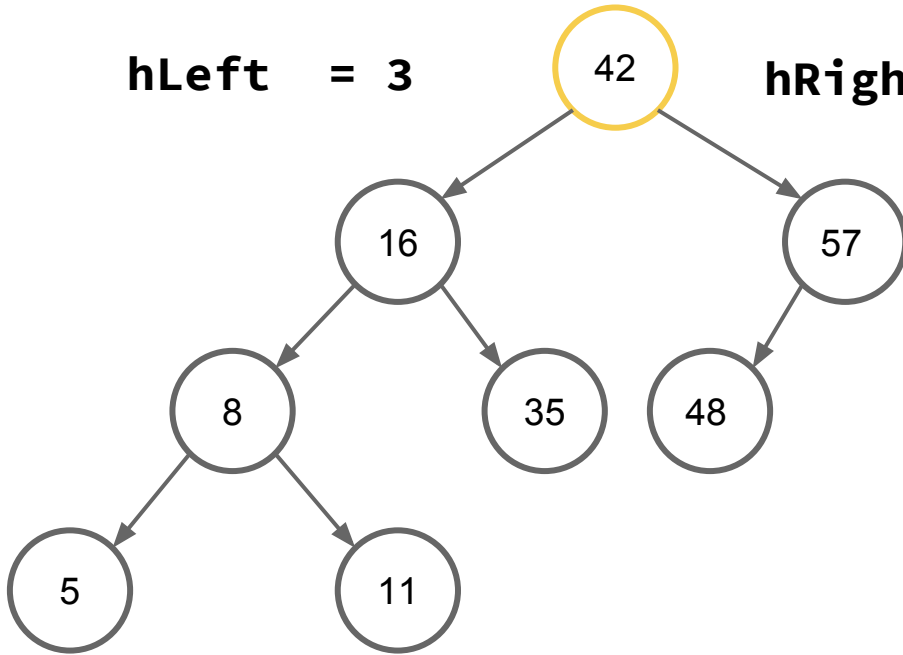
diferença

definição

1 = 1

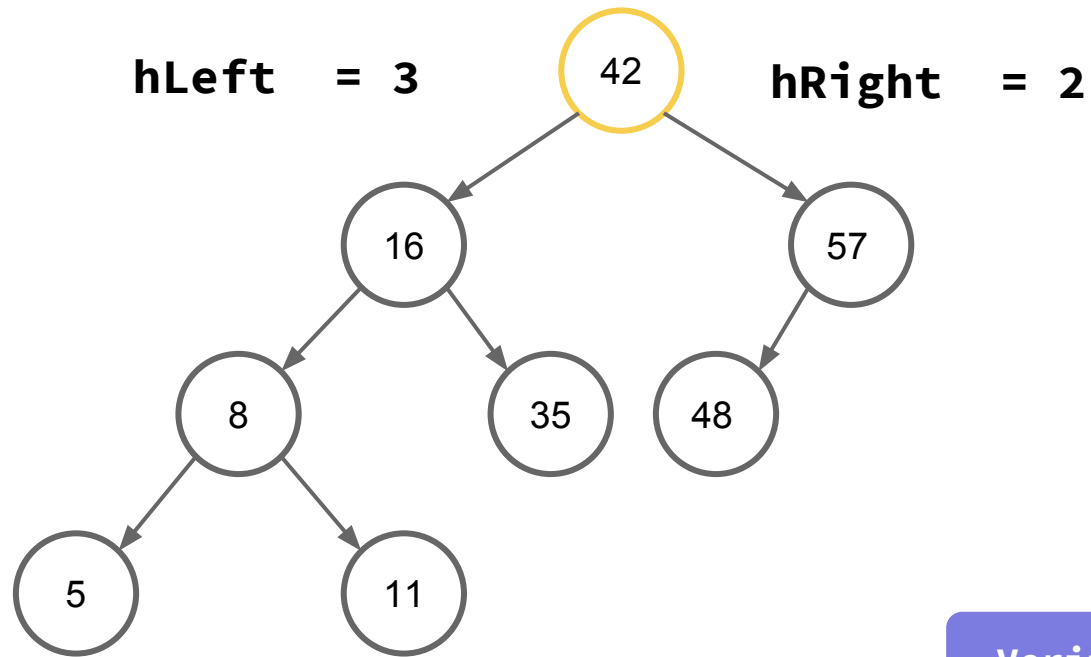
hLeft = 3

hRight = 2



Balanceada?

diferença de altura
das duas subárvores
de **qualquer** nó é
menor ou igual a 1



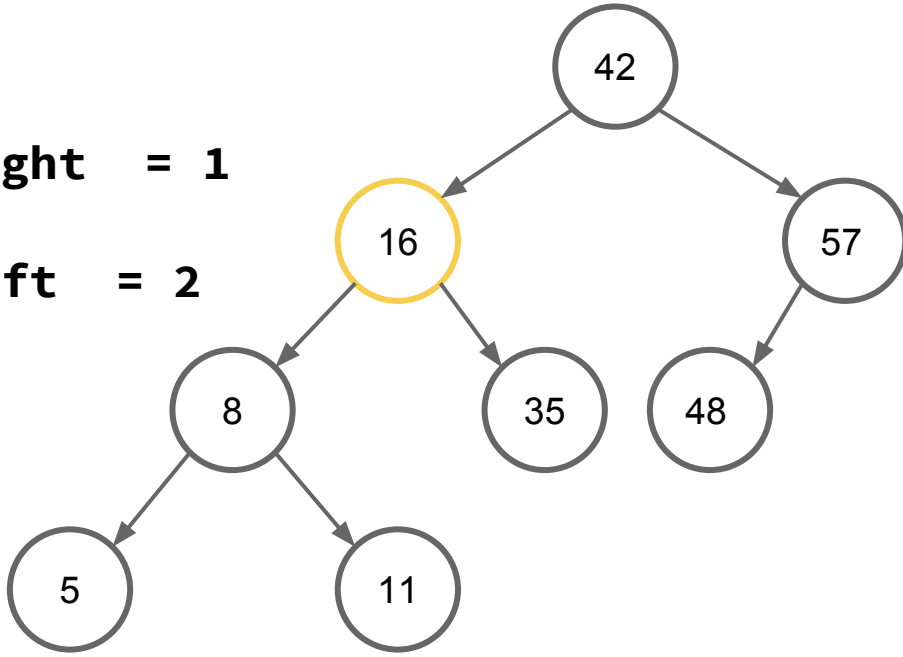
Balanced?

diferença de altura
das duas subárvores
de qualquer nó é
menor ou igual a 1

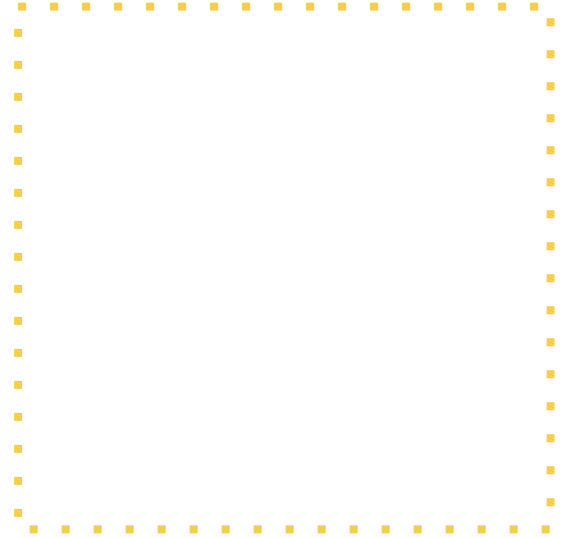
Verificar TODOS os nós

hRight = 1

hLeft = 2

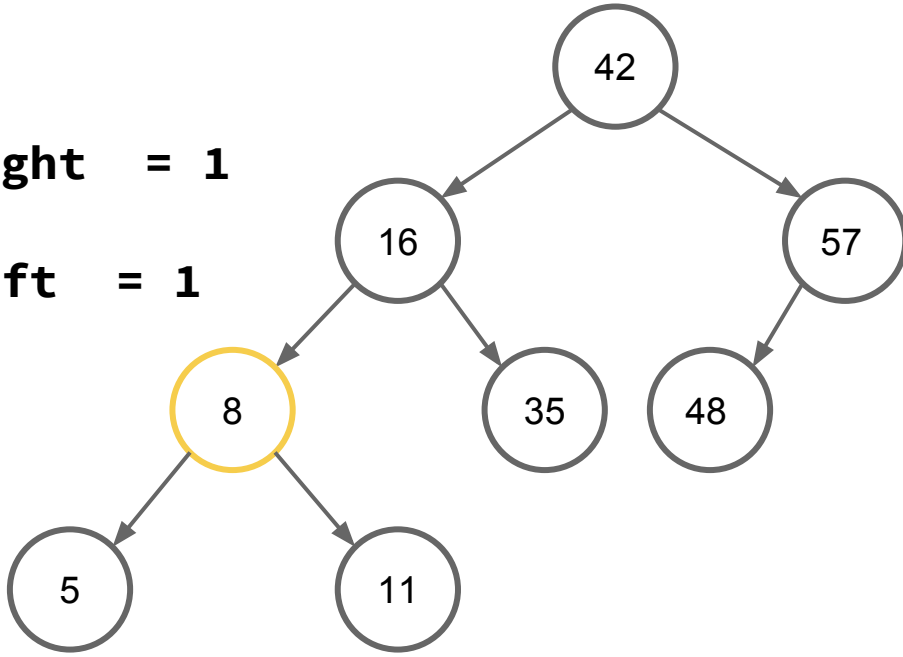


Balanceada?

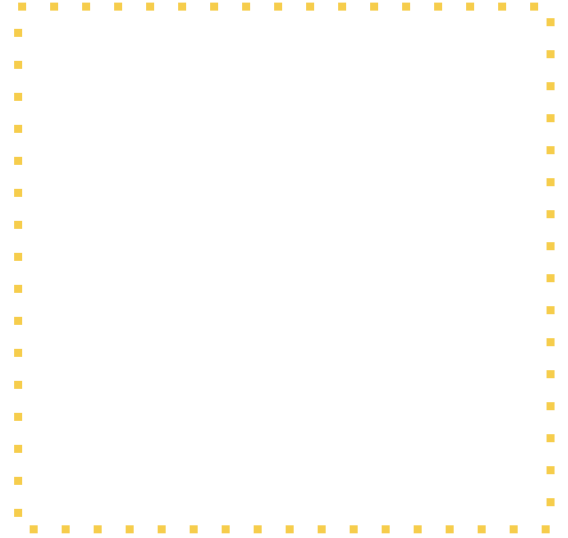


hRight = 1

hLeft = 1

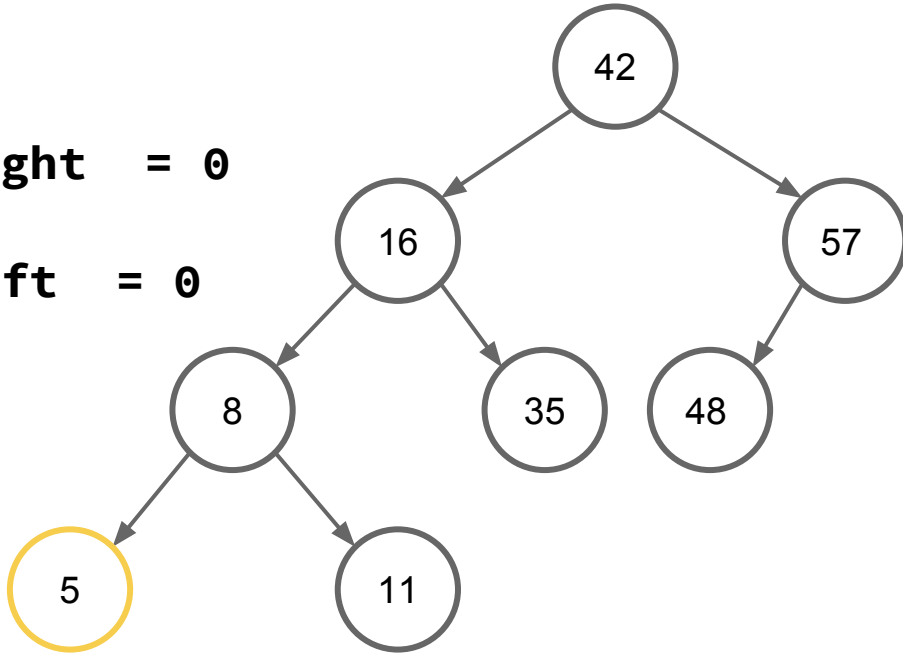


Balanceada?

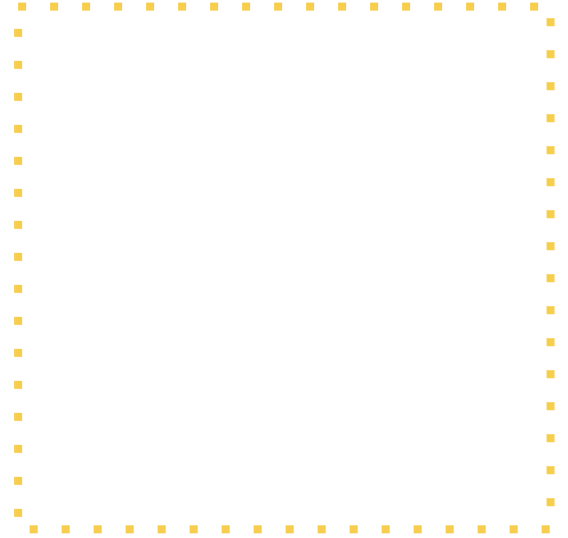


hRight = 0

hLeft = 0

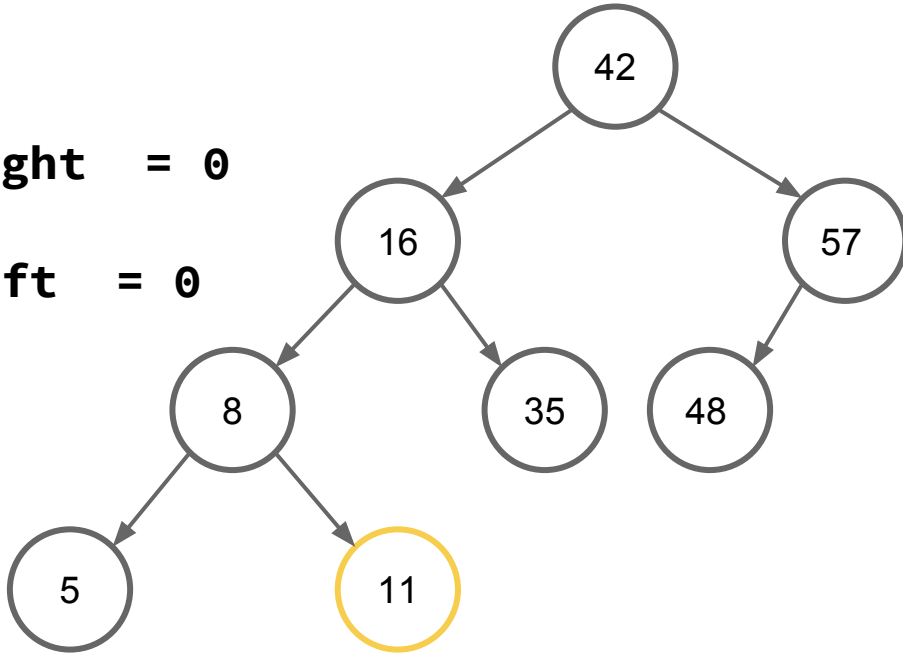


Balanceada?

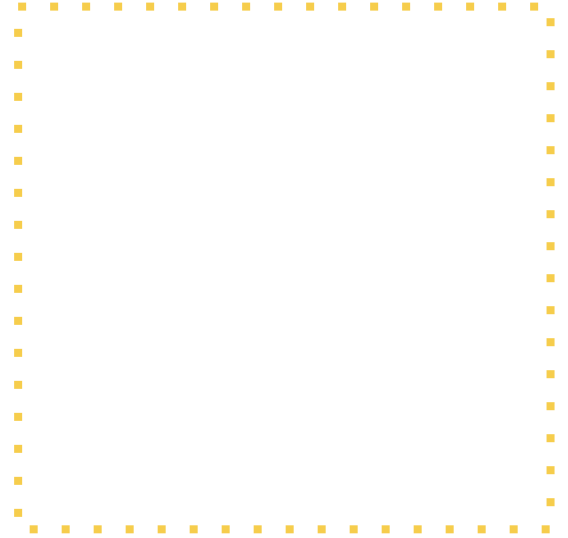


hRight = 0

hLeft = 0

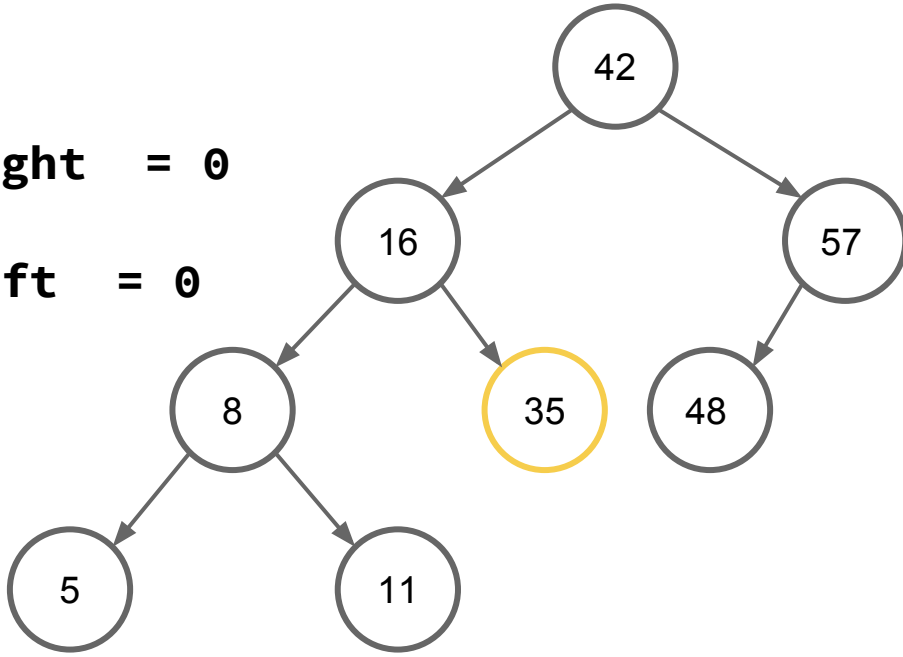


Balanceada?

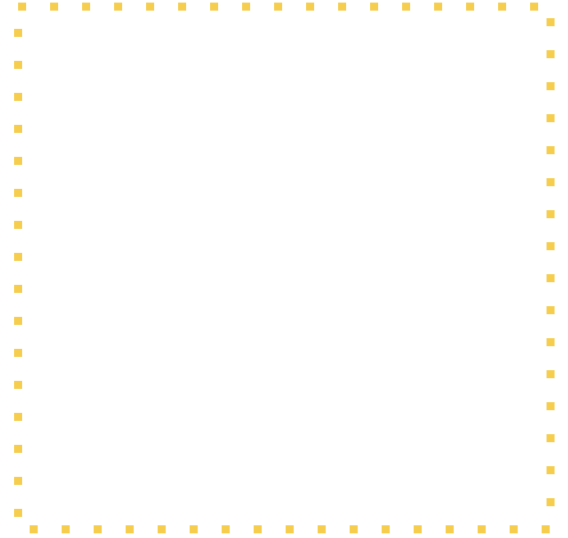


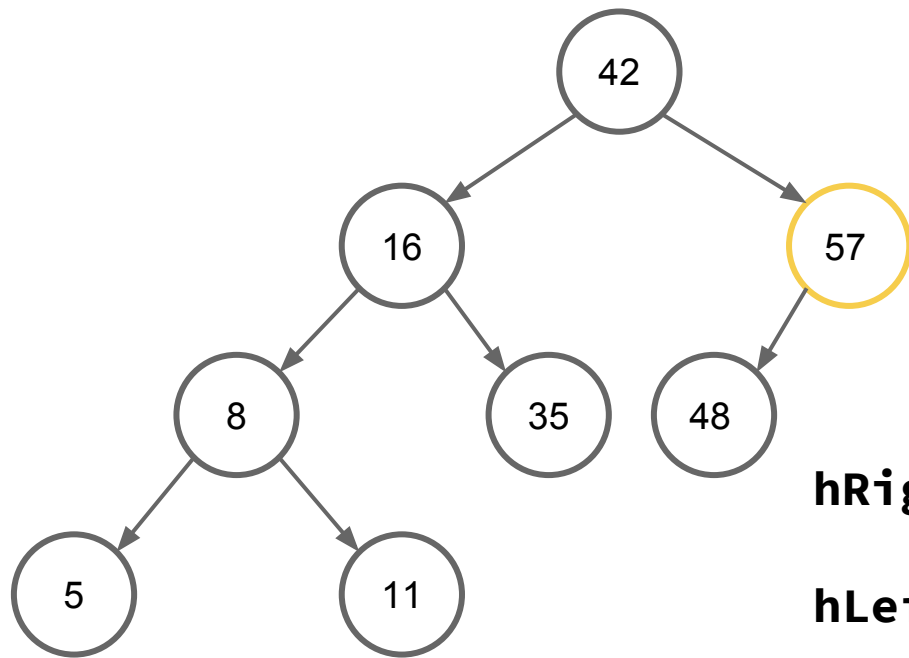
hRight = 0

hLeft = 0



Balanceada?

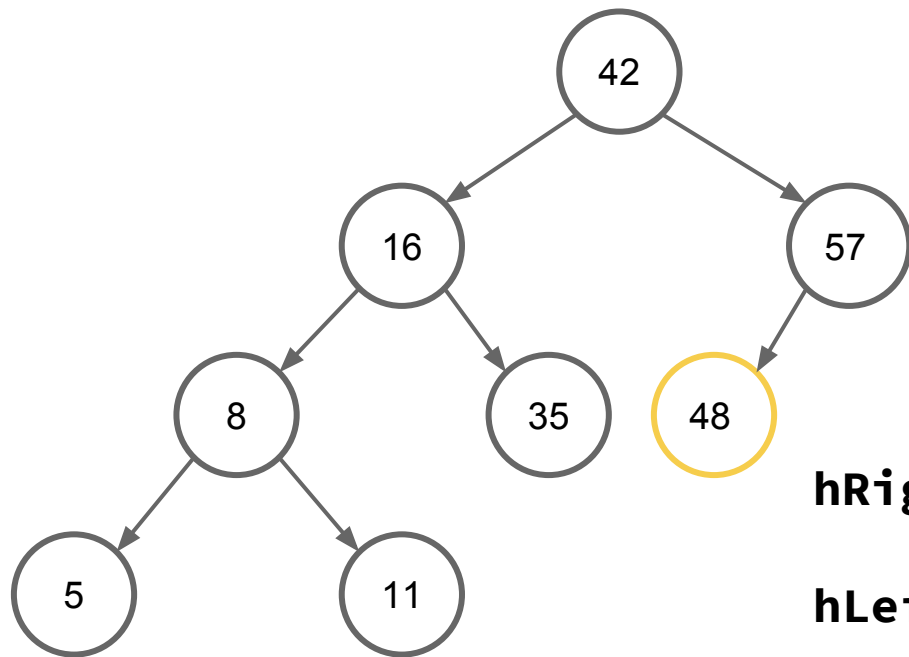




hRight = 0

hLeft = 1

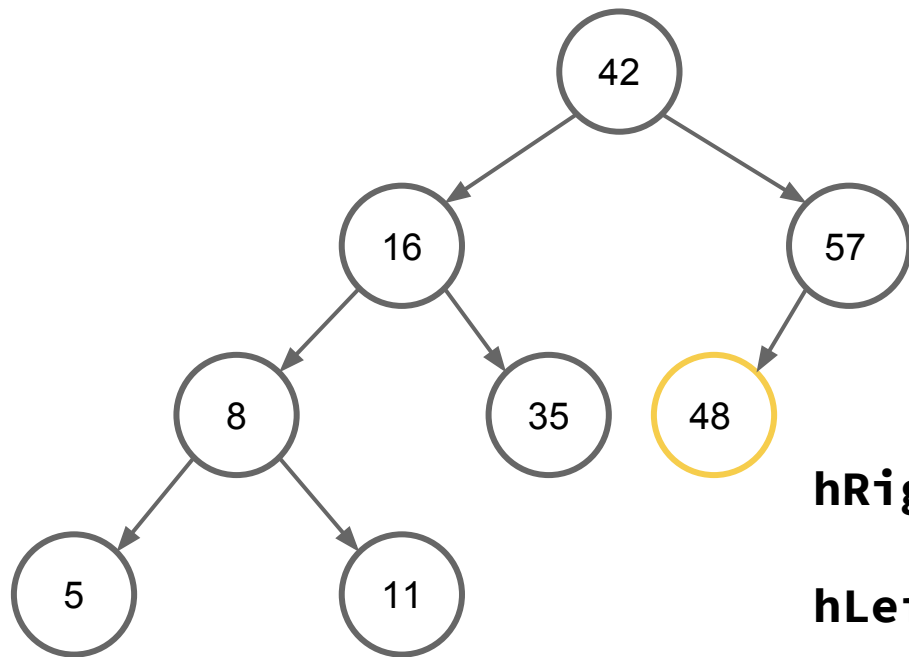
Balanceada?



hRight = 0

hLeft = 0

Balanceada?



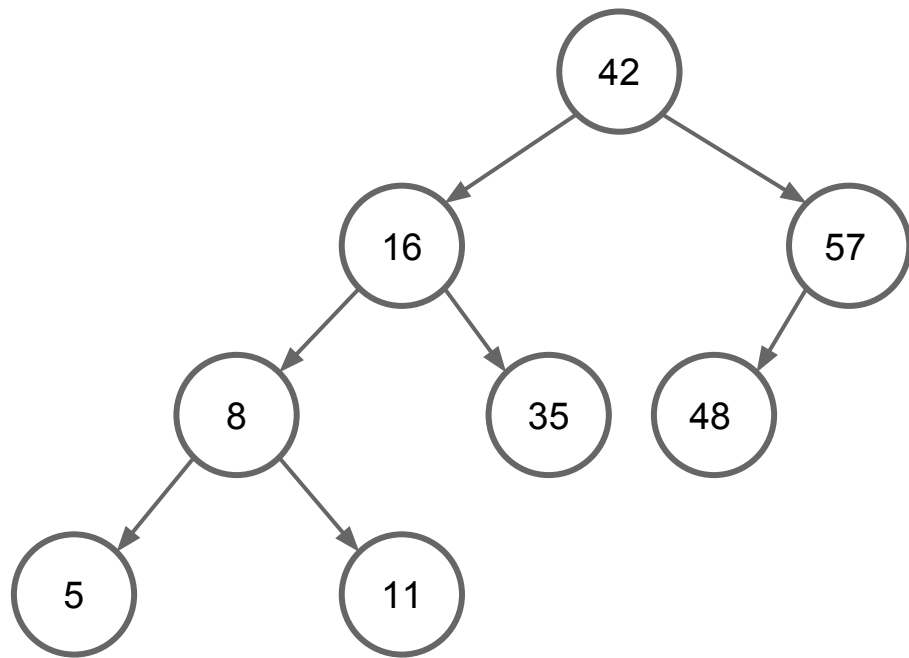
hRight = 0

hLeft = 0

Balanceada?

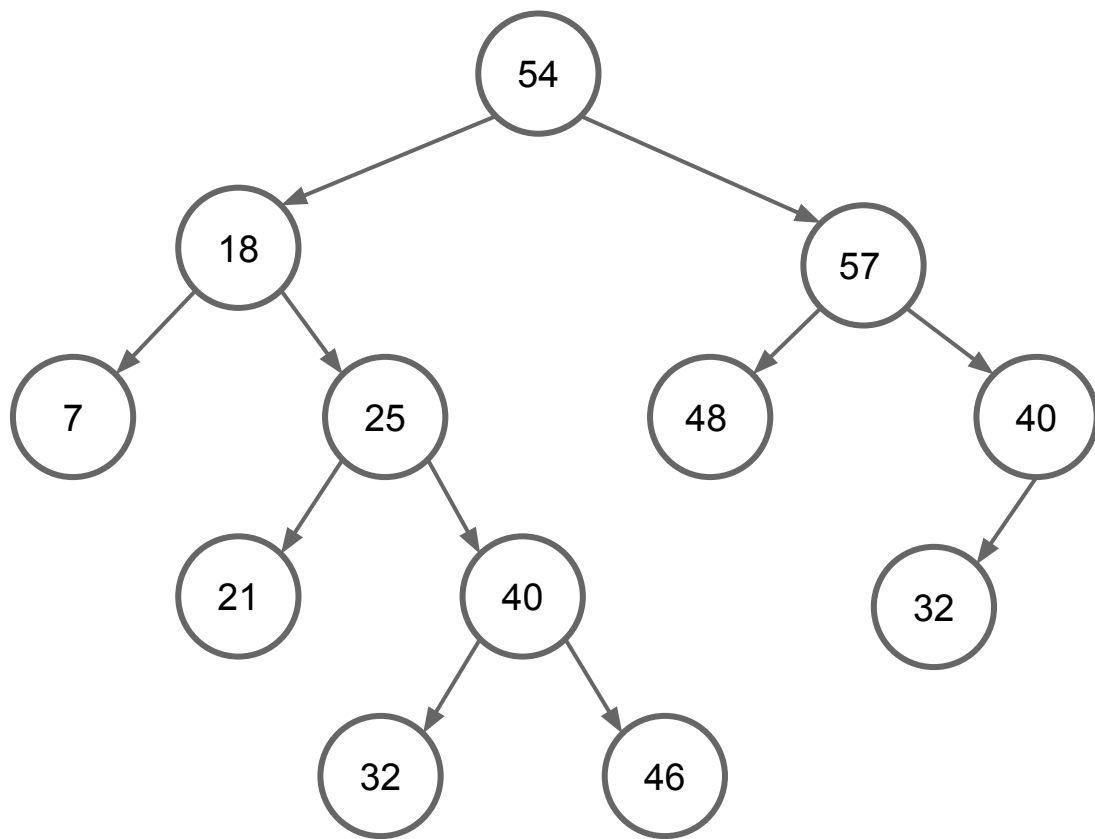
Sim!

diferença de altura
das duas subárvores
de qualquer nó é
menor ou igual a 1



Balanceada



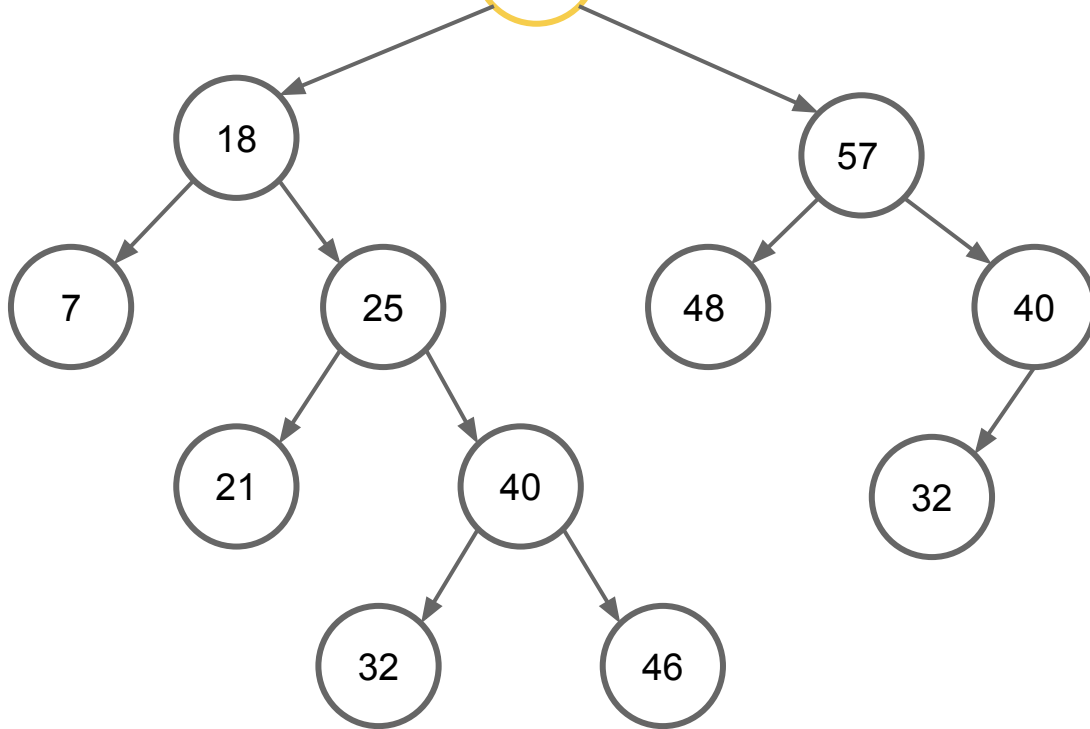


Balanceada?

hLeft = 4

54

hRight = 3

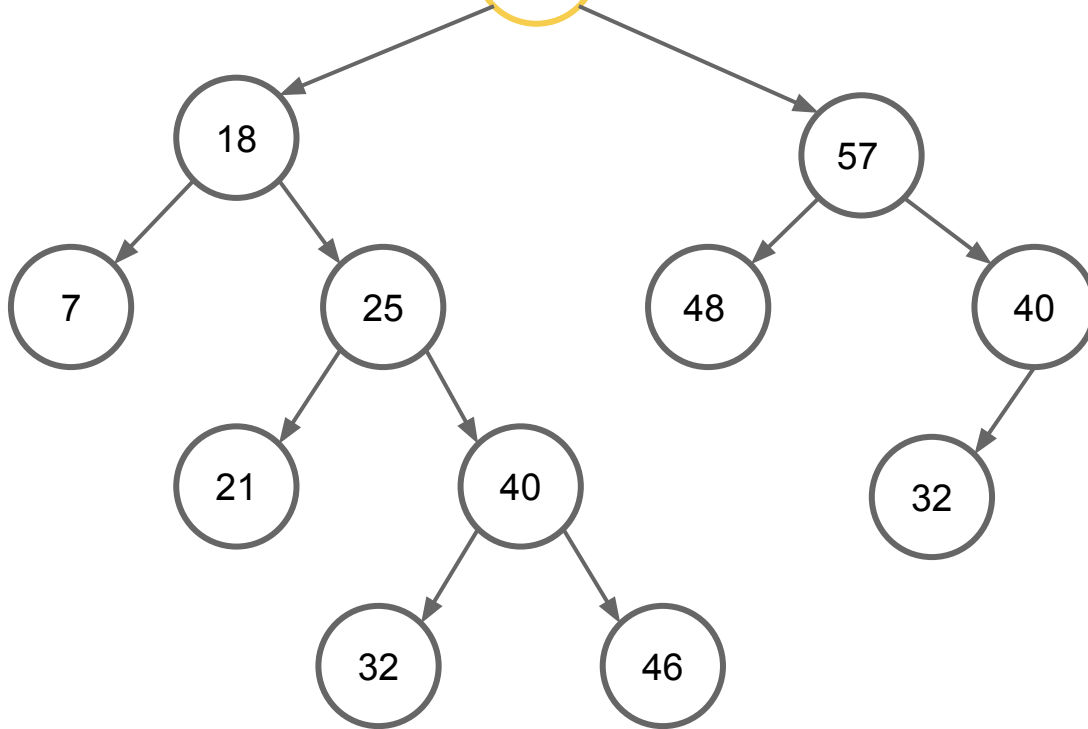


Balanceada?

hLeft = 4

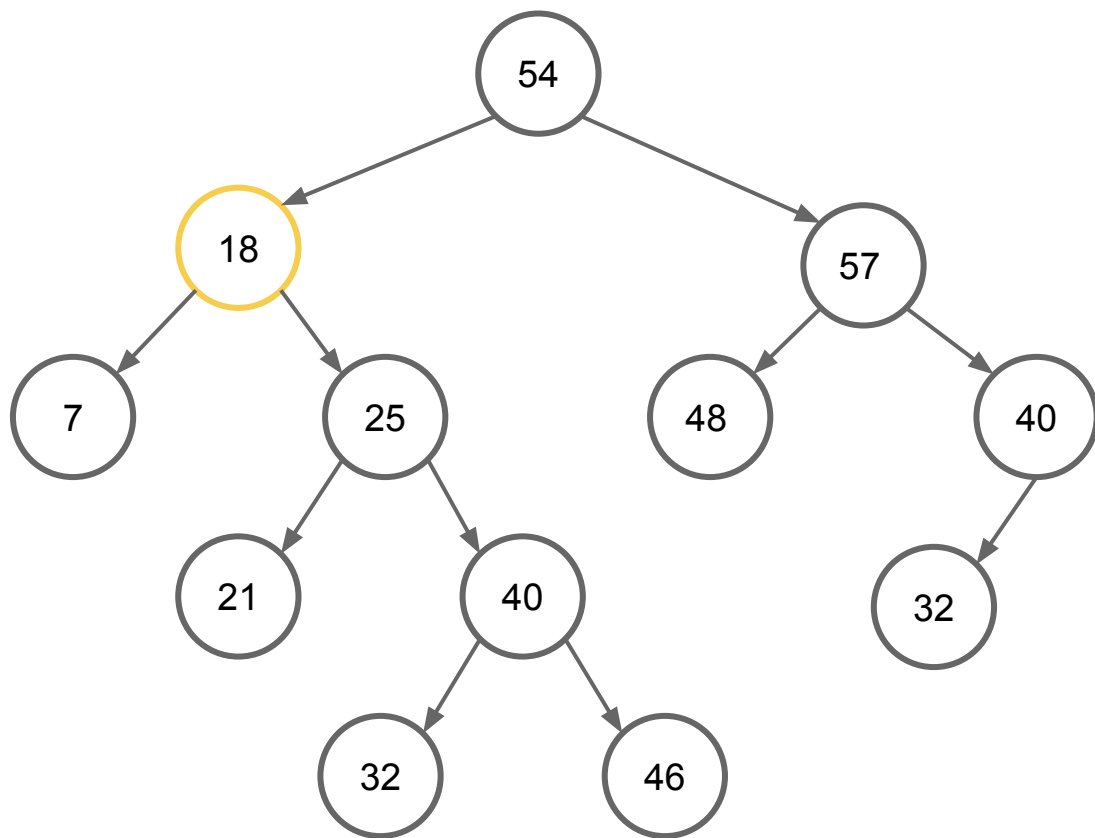
54

hRight = 3

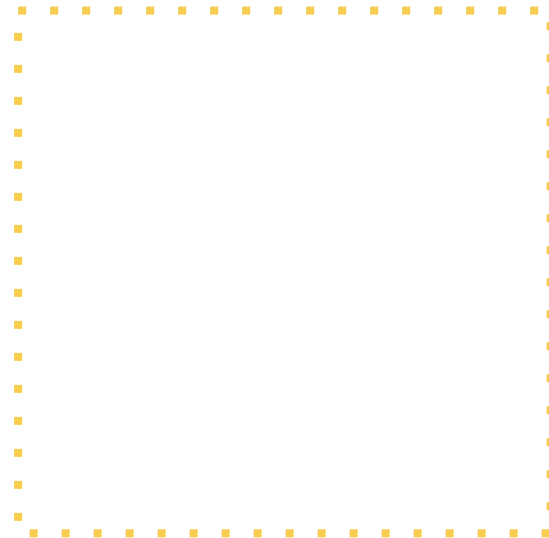


Balanceada?

1 = 1

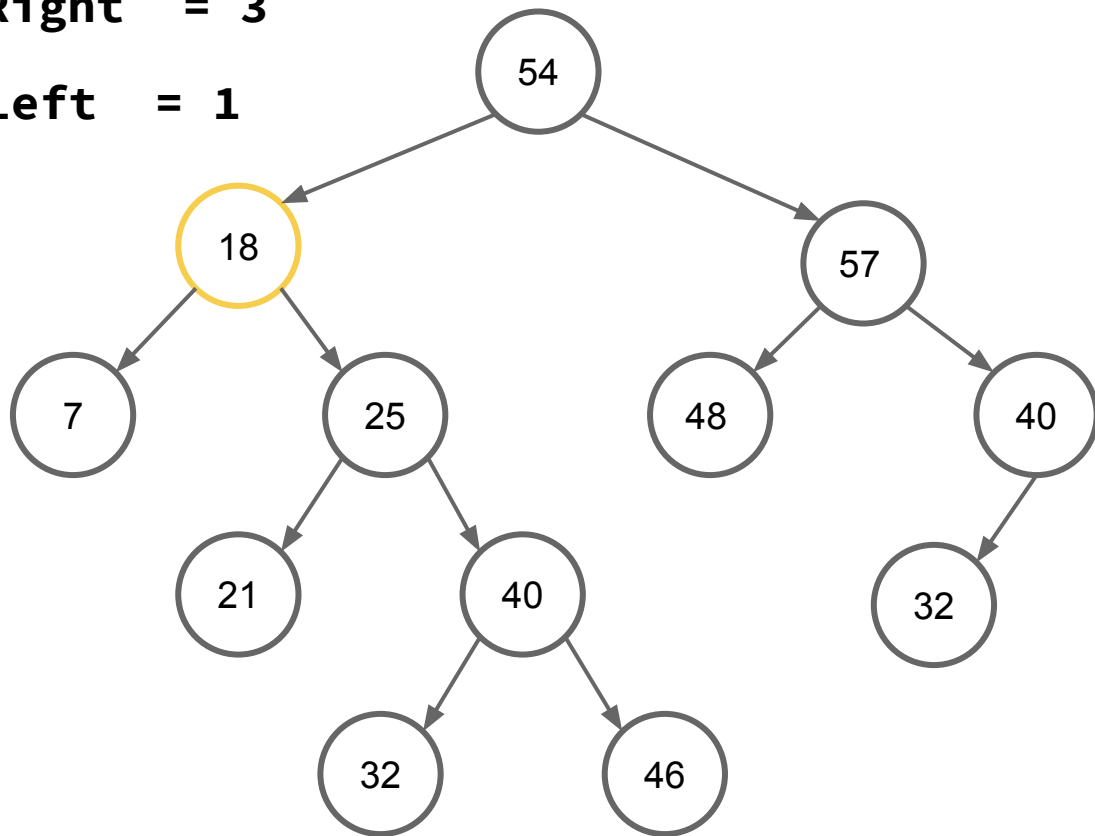


Balanceada?



hRight = 3

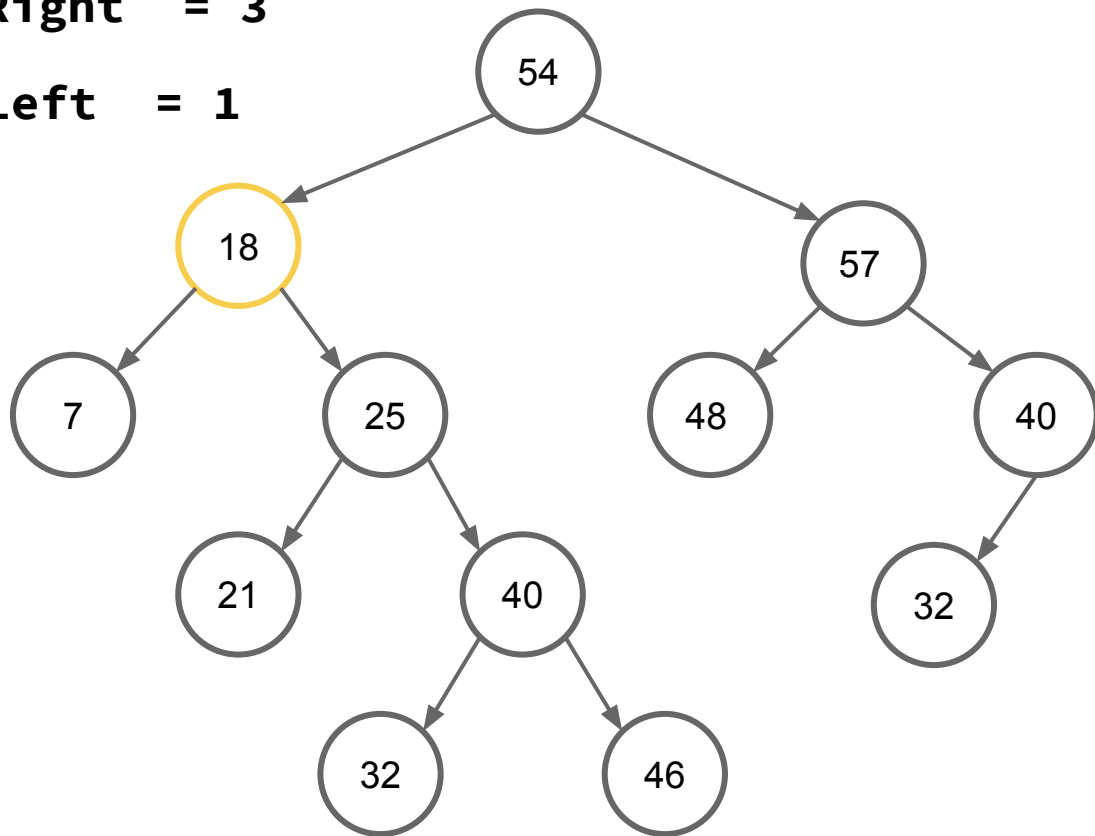
hLeft = 1



Balanceada?

hRight = 3

hLeft = 1



Desbalanceada

2 > 1

BALANCEADA?

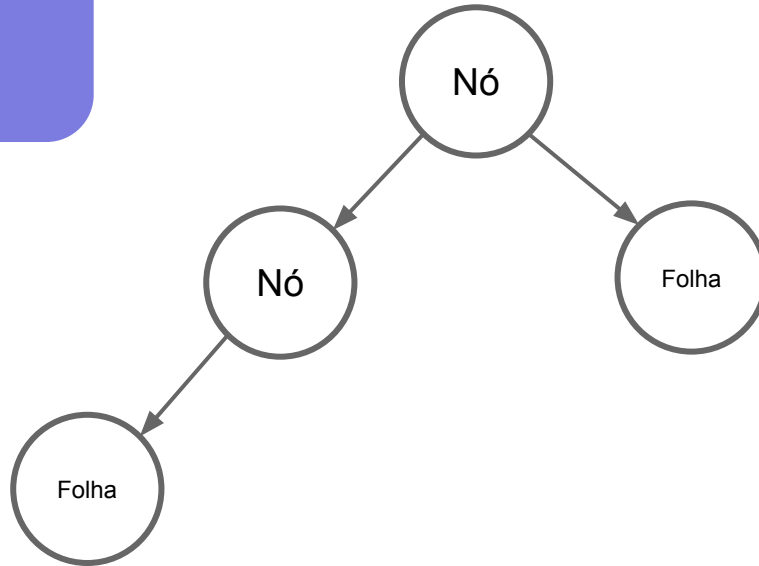
- 1) `no = raiz`
- 2) **SE** `no == NULL`:
 - a) Árvore vazia, altura é 0
- 3) **SE** `abs(esquerda.altura, direita.altura) > 1`
 - a) Desbalanceada
- 4) **SENÃO**
 - a) **SE** `balanceada(esquerda)` e `balanceada(direita)`
 - i) Balanceada

Verificar
todos os nós

Recursiva

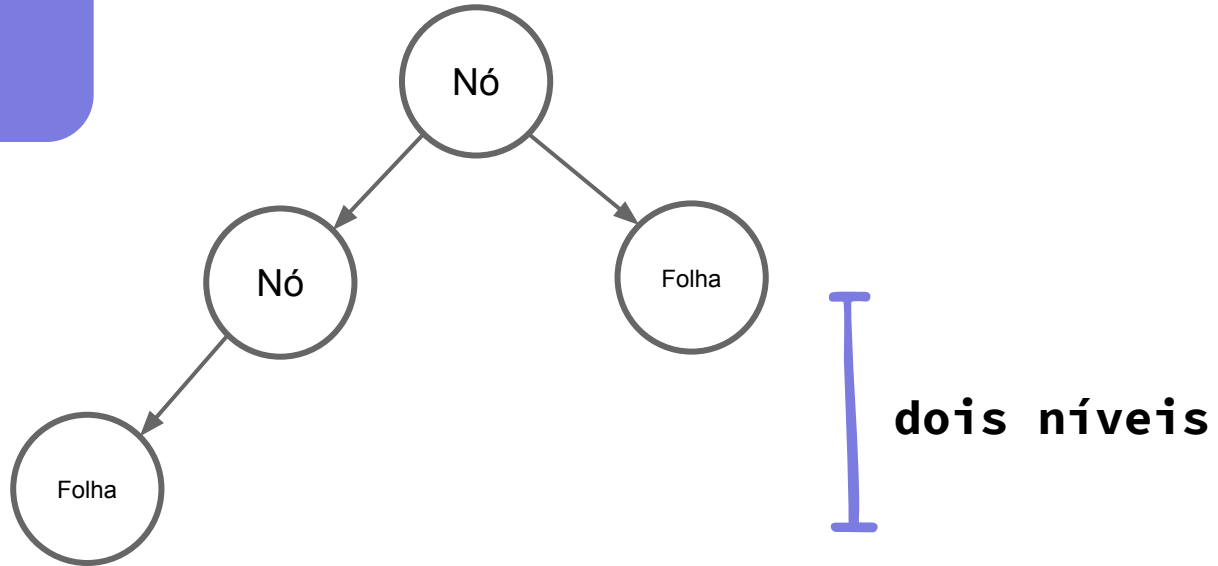
ÁRVORE PERFEITAMENTE BALANCEADA

As folhas estão em
no máximo dois
níveis distintos



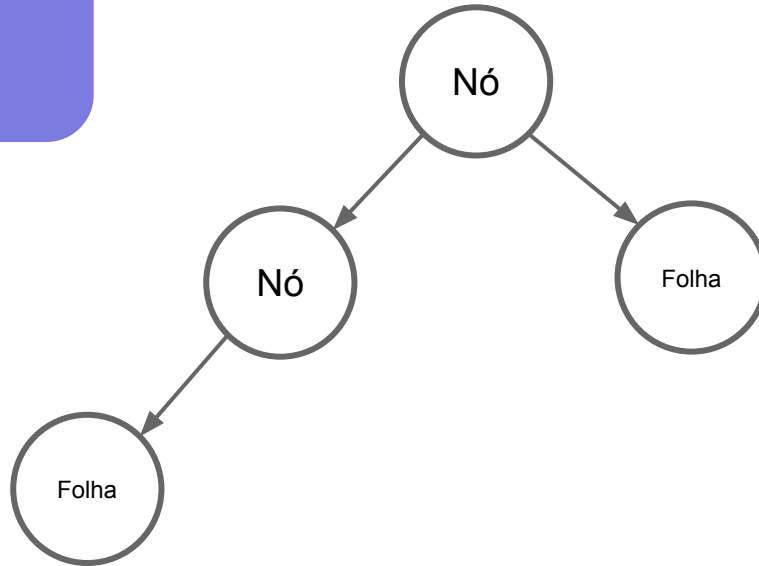
ÁRVORE PERFEITAMENTE BALANCEADA

As folhas estão em
no máximo dois
níveis distintos

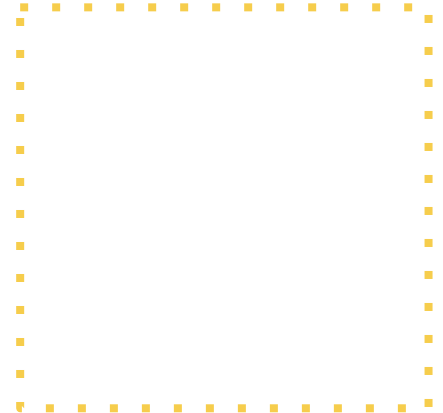


ÁRVORE PERFEITAMENTE BALANCEADA

As folhas estão em
no máximo dois
níveis distintos

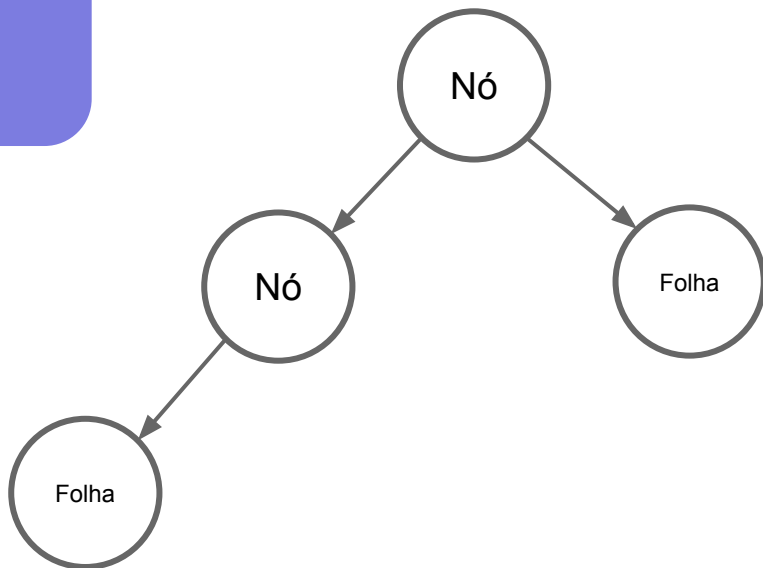


Balanceda?



ÁRVORE PERFEITAMENTE BALANCEADA

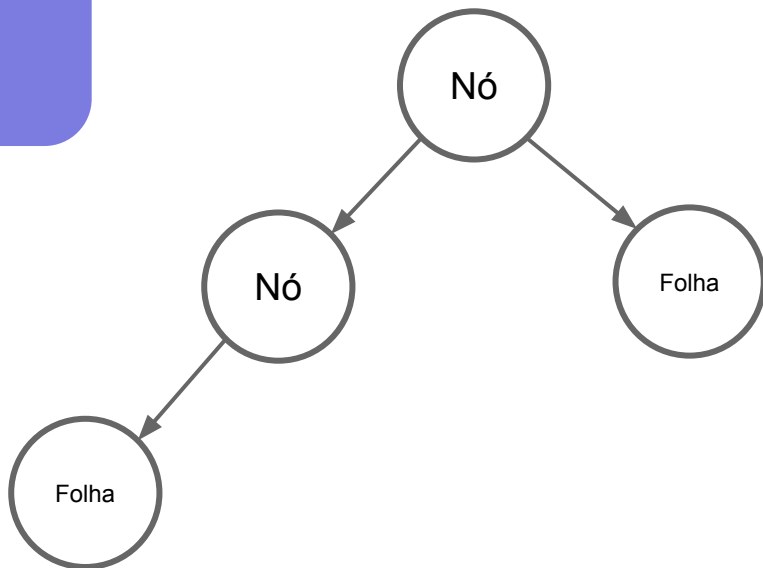
As folhas estão em
no máximo dois
níveis distintos



Balanceda

ÁRVORE PERFEITAMENTE BALANCEADA

As folhas estão em
no máximo dois
níveis distintos



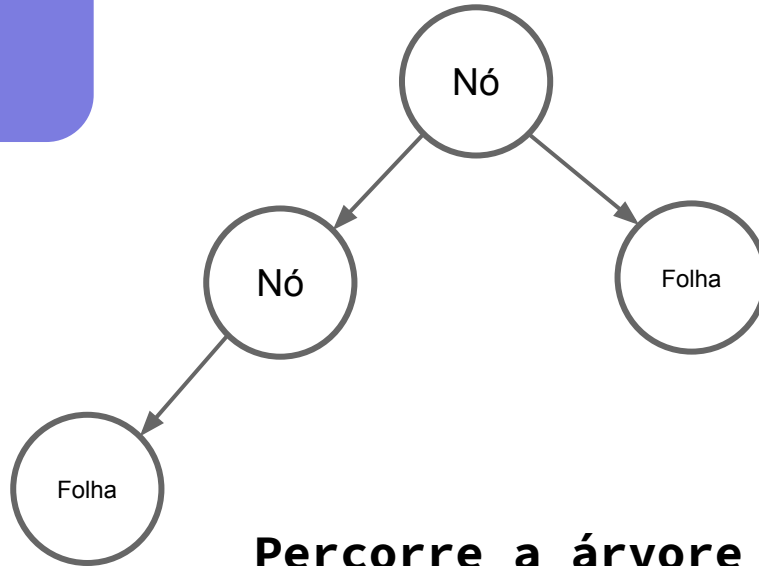
Balanceda

minLeaf = 0

maxLeaf = 0

ÁRVORE PERFEITAMENTE BALANCEADA

As folhas estão em
no máximo dois
níveis distintos



Percorre a árvore

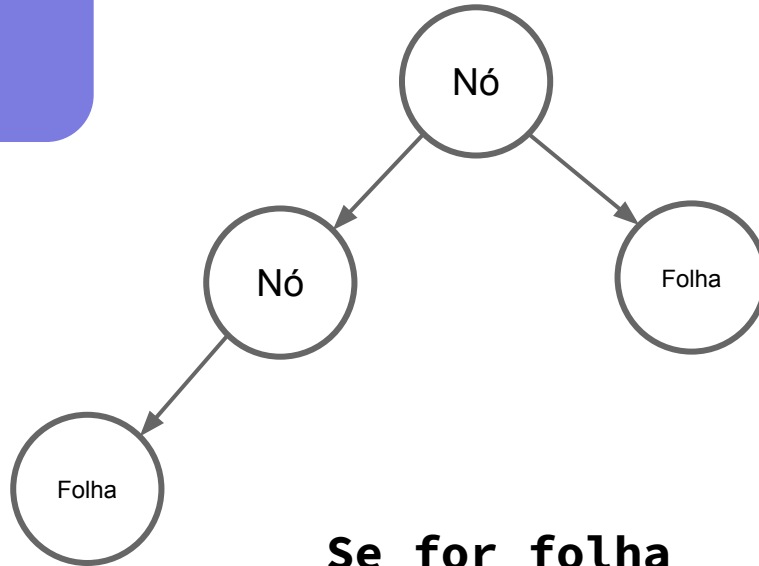
Balanceda

minLeaf = 0

maxLeaf = 0

ÁRVORE PERFEITAMENTE BALANCEADA

As folhas estão em
no máximo dois
níveis distintos



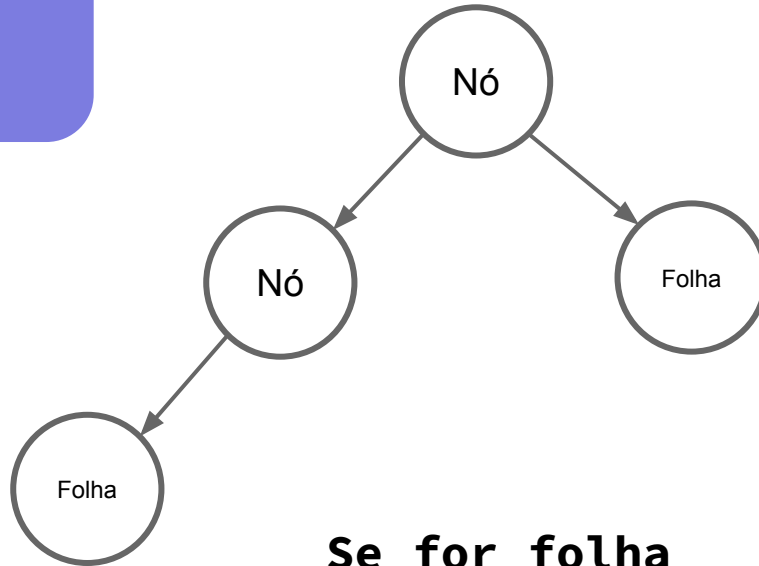
Balanceda

minLeaf = 0

maxLeaf = 0

ÁRVORE PERFEITAMENTE BALANCEADA

As folhas estão em
no máximo dois
níveis distintos



Se for folha
atualiza min e max

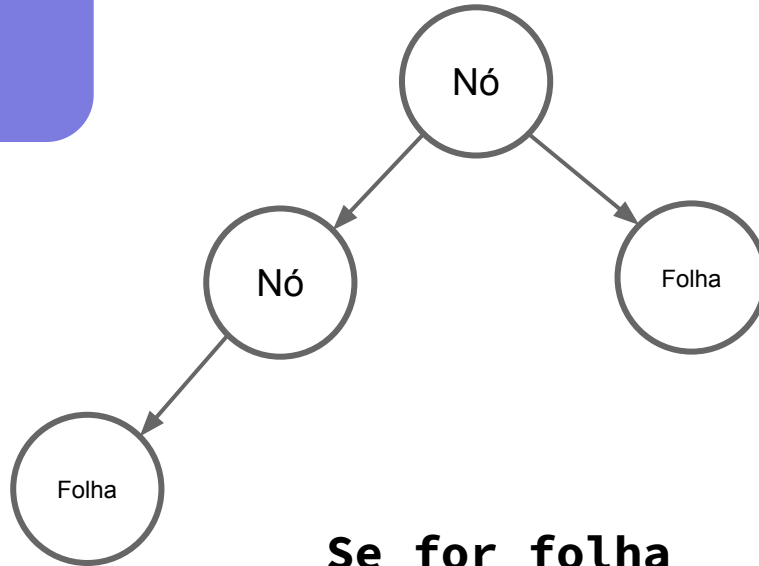
Balanceda

minLeaf = 0

maxLeaf = 0

ÁRVORE PERFEITAMENTE BALANCEADA

As folhas estão em
no máximo dois
níveis distintos



Se for folha
atualiza min e max

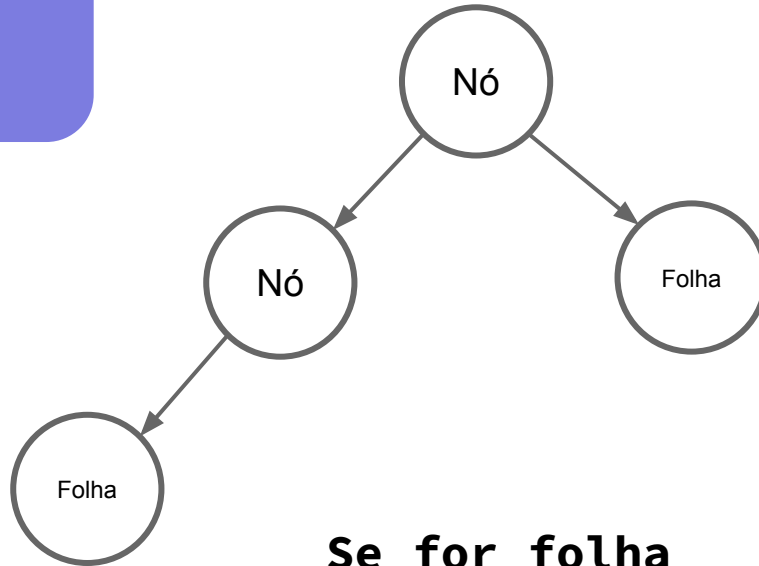
Balanceda

minLeaf = 1

maxLeaf = 2

ÁRVORE PERFEITAMENTE BALANCEADA

As folhas estão em
no máximo dois
níveis distintos



**Se for folha
atualiza min e max**

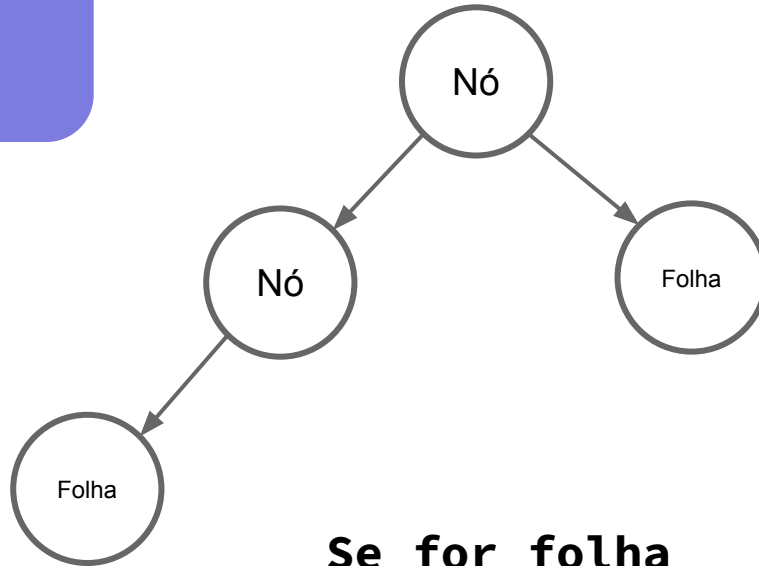
**Perfeitamente
Balanceada**

minLeaf = 1

maxLeaf = 2

ÁRVORE PERFEITAMENTE BALANCEADA

As folhas estão em
no máximo dois
níveis distintos



**Se for folha
atualiza min e max**

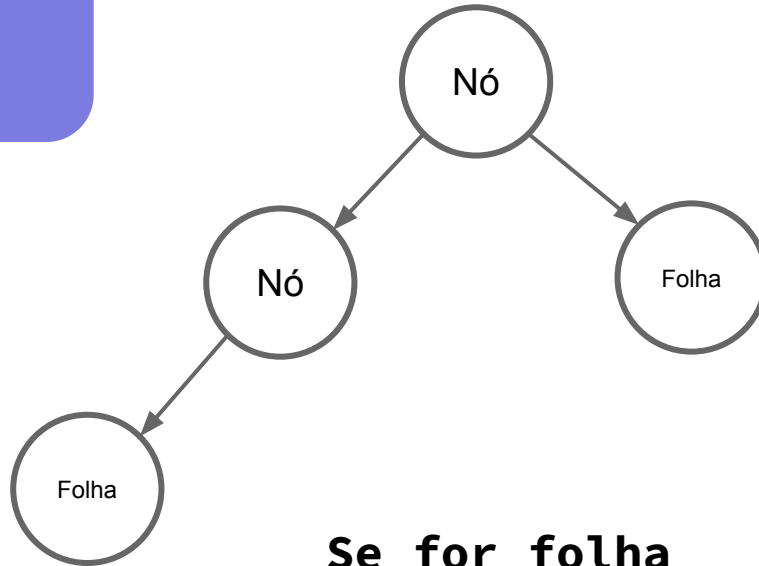
Balanceda

minLeaf = 1

maxLeaf = 3

ÁRVORE PERFEITAMENTE BALANCEADA

As folhas estão em
no máximo dois
níveis distintos



**Se for folha
atualiza min e max**

**!Perfeitamente
Balanceada**

minLeaf = 1

maxLeaf = 3

O DESBALANCEAMENTO PREJUDICA A BUSCA

Pode virar $O(n)$

O DESBALANCEAMENTO PRETUDICA BUSCA

Pode virar

Como balancear
a árvore?

OPERAÇÕES

Inserção ✓

Busca ✓

Remoção ✓

Travessia ✓

Balanceamento

INSERÇÃO

Inserção controlada

Ordenar os valores

Inserir o médio na árvore

Repete com a parte esquerda

Repete com a parte direita

Busca binária

ROTAÇÃO

Rotacionar desbalanceada

Aplica rotações

Transforma em backbone

Aplica rotações

Rotaciona até balancear

Magia?

BALANCEAMENTO



INSERÇÃO

Inserção controlada

Ordenar os valores

Inserir o médio na árvore

Repete com a parte esquerda

Repete com a parte direita

INSERÇÃO

Inserção controlada

Ordenar os valores

Inserir o médio na árvore

Repete com a parte esquerda

Repete com a parte direita



In
order

Inserção

INSERÇÃO

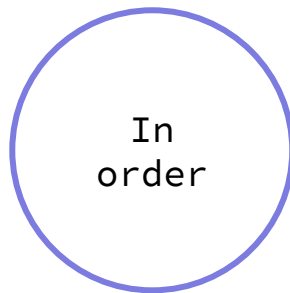
Inserção controlada

Ordenar os valores

Inserir o médio na árvore

Repete com a parte esquerda

Repete com a parte direita



Inserção

INSERÇÃO

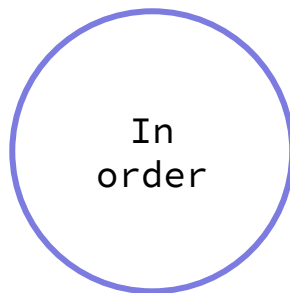
Inserção controlada

Ordenar os valores

Inserir o médio na árvore

Repete com a parte esquerda

Repete com a parte direita



Inserção

INSERÇÃO

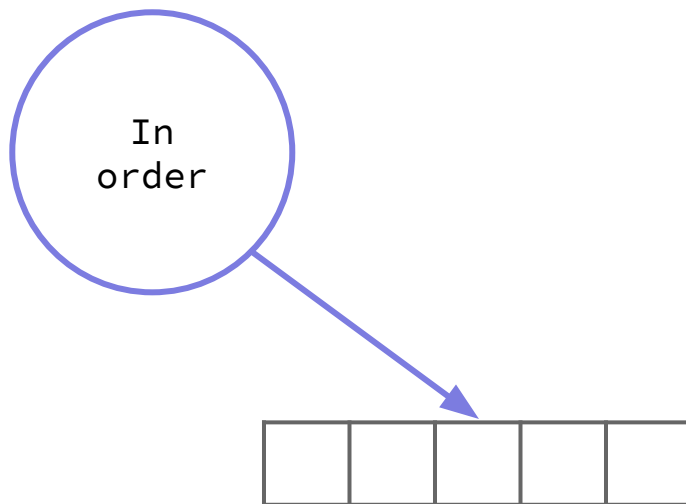
Inserção controlada

Ordenar os valores

Inserir o médio na árvore

Repete com a parte esquerda

Repete com a parte direita

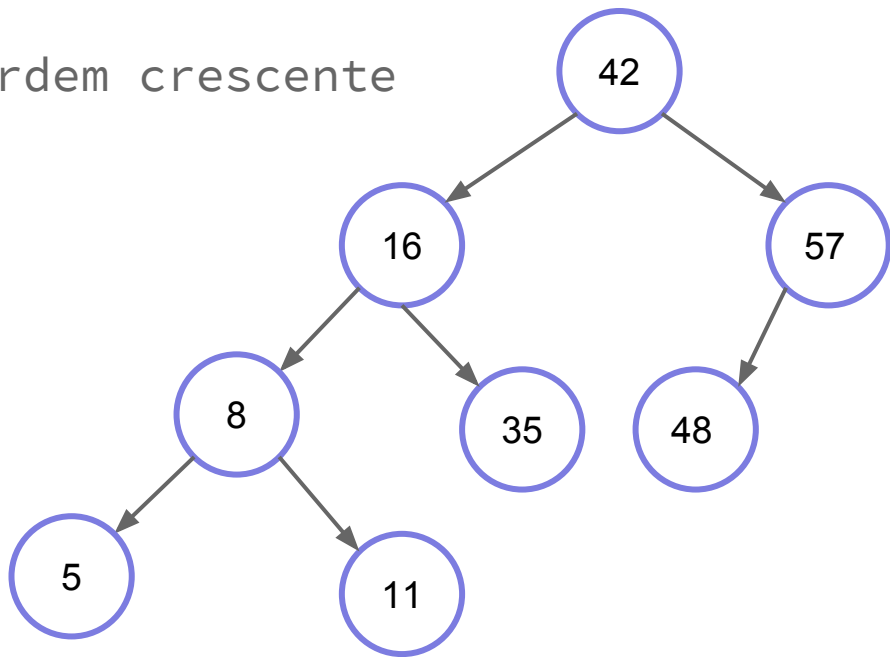


Inserção

TRAVESSIA IN-ORDER

- Percorrer a árvore: ordem crescente

5, 8, 11, 16, 35,
42, 48, 57



INSERÇÃO

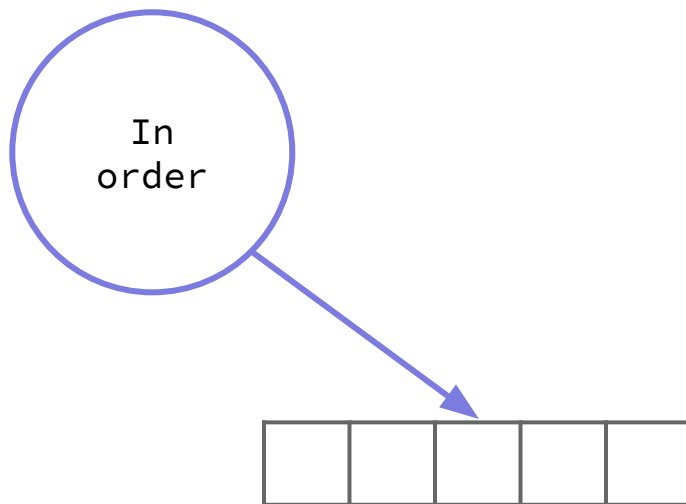
Inserção controlada

Ordenar os valores

Inserir o médio na árvore

Repete com a parte esquerda

Repete com a parte direita



Inserção

INSERÇÃO

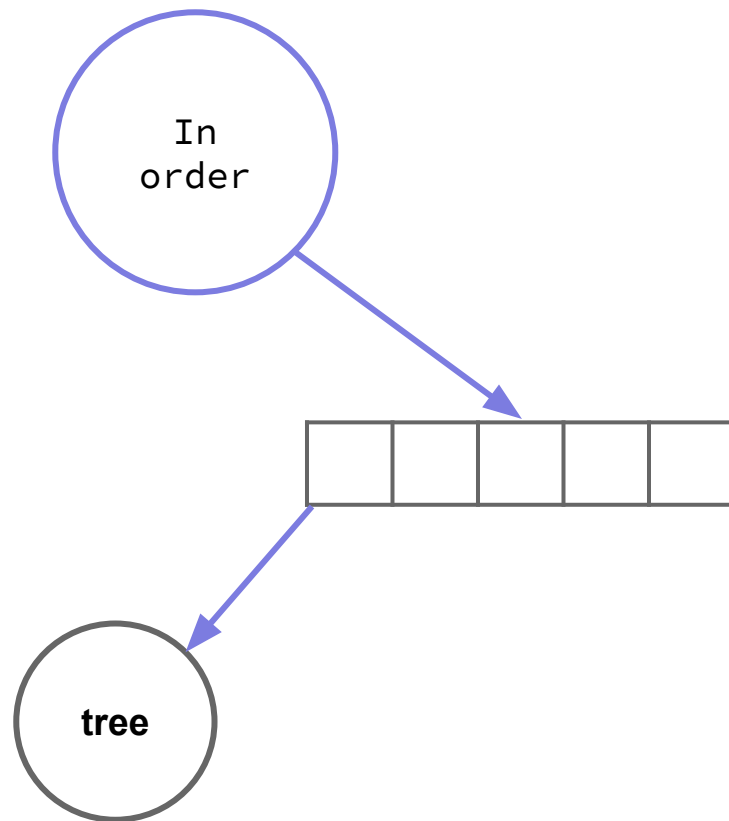
Inserção controlada

Ordenar os valores

Inserir o médio na árvore

Repete com a parte esquerda

Repete com a parte direita



Inserção

INSERÇÃO



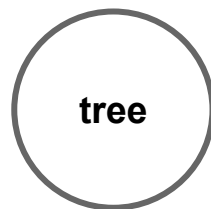
Inserção controlada

Ordenar os valores

Inserir o médio na árvore

Repete com a parte esquerda

Repete com a parte direita



Inserção

INSERÇÃO

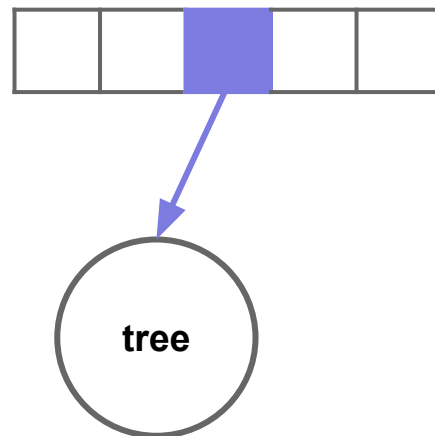
Inserção controlada

Ordenar os valores

Inserir o médio na árvore

Repete com a parte esquerda

Repete com a parte direita



Inserção

INSERÇÃO

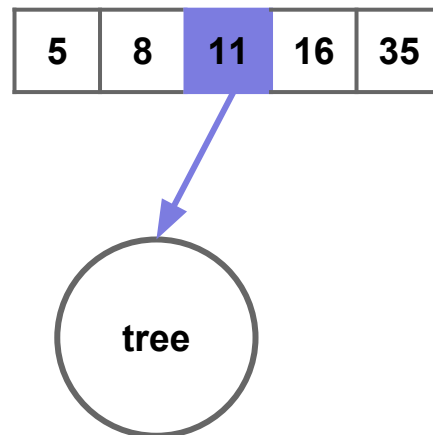
Inserção controlada

Ordenar os valores

Inserir o médio na árvore

Repete com a parte esquerda

Repete com a parte direita



Inserção

INSERÇÃO

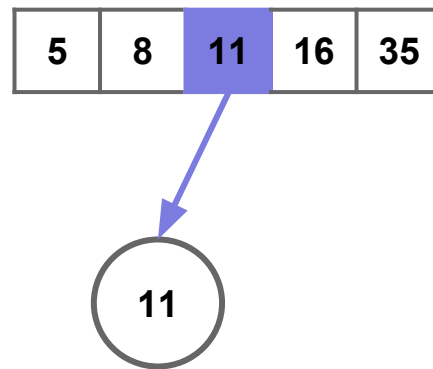
Inserção controlada

Ordenar os valores

Inserir o médio na árvore

Repete com a parte esquerda

Repete com a parte direita



Inserção

INSERÇÃO

Inserção controlada

Ordenar os valores

Inserir o médio na árvore

Repete com a parte esquerda

Repete com a parte direita

5	8	11	16	35
---	---	----	----	----



Inserção

INSERÇÃO

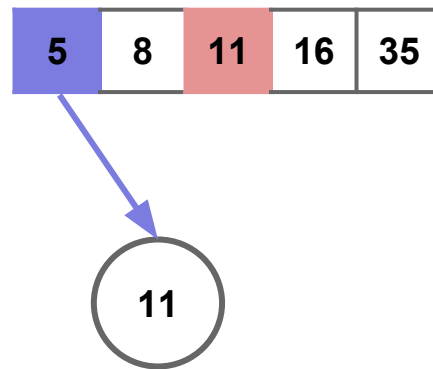
Inserção controlada

Ordenar os valores

Inserir o médio na árvore

Repete com a parte esquerda

Repete com a parte direita



Inserção

INSERÇÃO

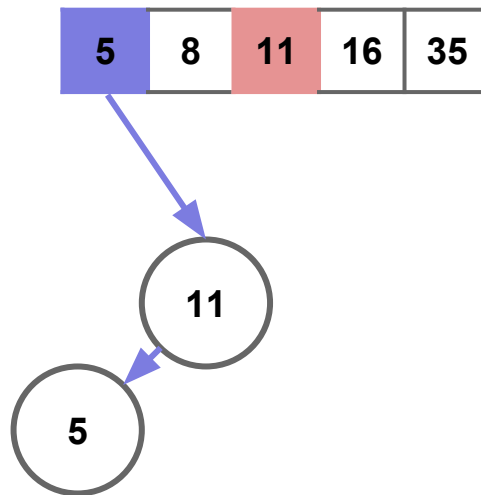
Inserção controlada

Ordenar os valores

Inserir o médio na árvore

Repete com a parte esquerda

Repete com a parte direita



Inserção

INSERÇÃO

Inserção controlada

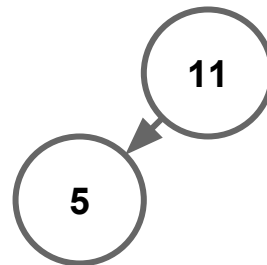
Ordenar os valores

Inserir o médio na árvore

Repete com a parte esquerda

Repete com a parte direita

5	8	11	16	35
---	---	----	----	----



INSERÇÃO

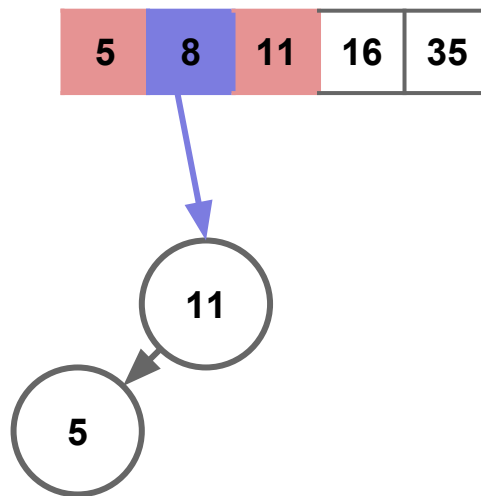
Inserção controlada

Ordenar os valores

Inserir o médio na árvore

Repete com a parte esquerda

Repete com a parte direita



Inserção

INSERÇÃO

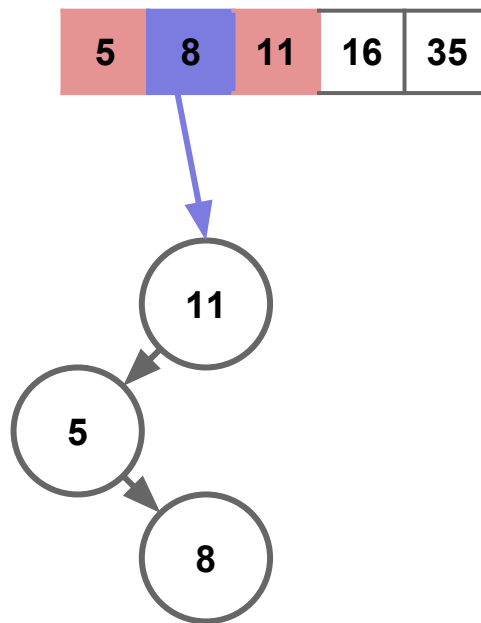
Inserção controlada

Ordenar os valores

Inserir o médio na árvore

Repete com a parte esquerda

Repete com a parte direita



Inserção

INSERÇÃO

5	8	11	16	35
---	---	----	----	----

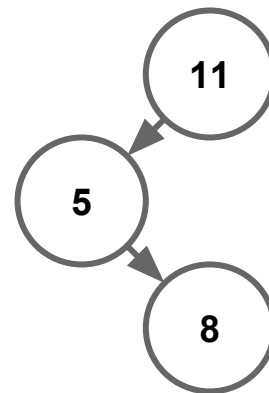
Inserção controlada

Ordenar os valores

Inserir o médio na árvore

Repete com a parte esquerda

Repete com a parte direita



Inserção

INSERÇÃO

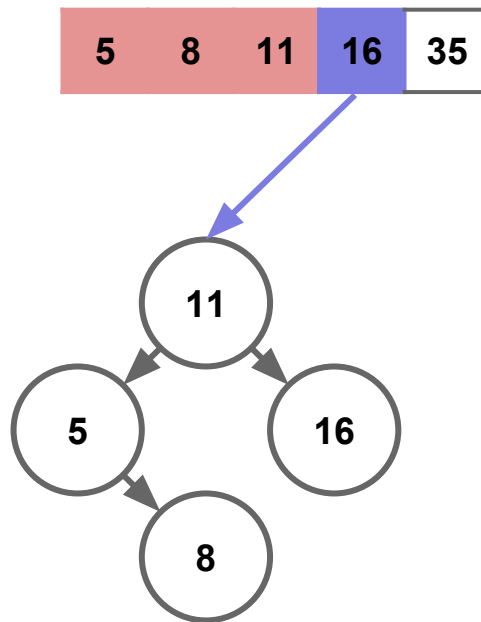
Inserção controlada

Ordenar os valores

Inserir o médio na árvore

Repete com a parte esquerda

Repete com a parte direita



Inserção

INSERÇÃO

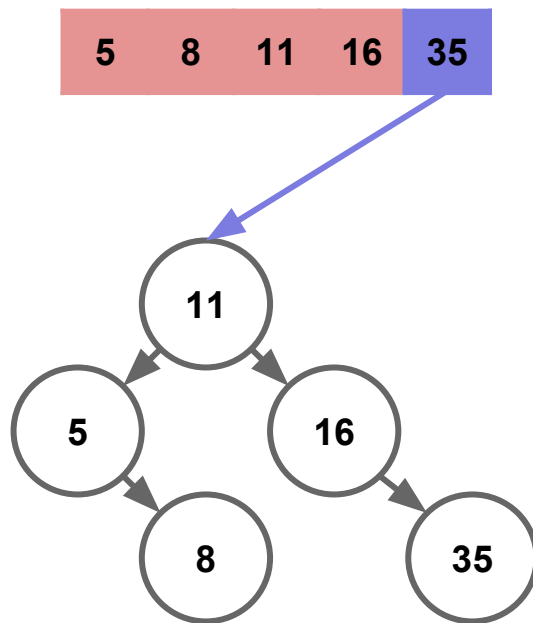
Inserção controlada

Ordenar os valores

Inserir o médio na árvore

Repete com a parte esquerda

Repete com a parte direita



Inserção

INSERÇÃO

Inserção controlada

Ordenar os valores

Inserir o médio na árvore

Repete com a parte esquerda

Repete com a parte direita

Busca binária

ROTAÇÃO

Rotacionar desbalanceada

Aplica rotações

Transforma em backbone

Aplica rotações

Rotaciona até balancear

Magia?

Árvore vazia

INSERÇÃO

- Requer memória(n)
 - Ordenação (count, merge...)
- + Insere na árvore em ordem

Busca binária

ROTAÇÃO

Rotacionar desbalanceada

Aplica rotações

Transforma em backbone

Aplica rotações

Rotaciona até balancear

Magia?

INSERÇÃO

- Requer memória(n)
 - Ordenação (count, merge...)
- + Insere na árvore em ordem

Busca binária

ROTAÇÃO

Rotacionar desbalanceada

Aplica rotações

Transforma em backbone

Aplica rotações

Rotaciona até balancear

Magia?

Árvore preenchida

INSERÇÃO

- + Ordenação $O(n)$
- Requer memória(n)
- Destruir e reconstruir

Busca binária

ROTAÇÃO

Rotacionar desbalanceada

Aplica rotações

Transforma em backbone

Aplica rotações

Rotaciona até balancear

Magia?

INSERÇÃO

- + Ordenação $O(n)$
- Requer memória(n)
- Destruir e reconstruir

Busca binária

ROTAÇÃO

Rotacionar desbalanceada

Aplica rotações

Transforma em backbone

Aplica rotações

Rotaciona até balancear

Magia?

ROTAÇÃO

Rotacionar desbalanceada

Aplica rotações

Transforma em backbone

Aplica rotações

Rotaciona até balancear

ROTAÇÃO

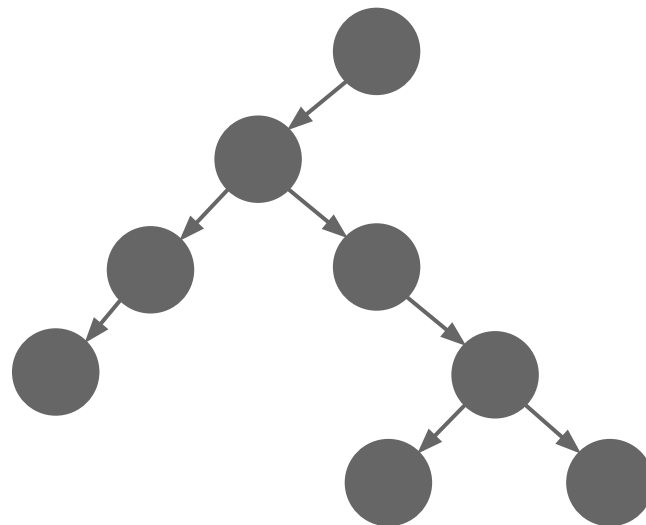
Rotacionar **desbalanceada**

Aplica rotações

Transforma em backbone

Aplica rotações

Rotaciona até balancear



Rotação

ROTAÇÃO

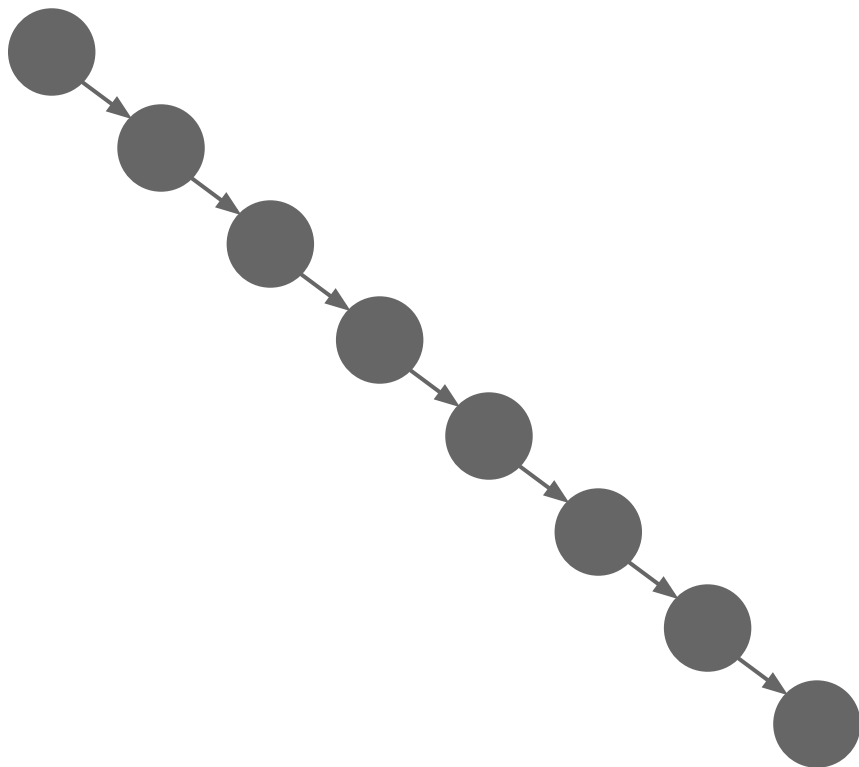
Rotacionar desbalanceada

Aplica rotações

Transforma em backbone

Aplica rotações

Rotaciona até balancear



Rotação

ROTAÇÃO

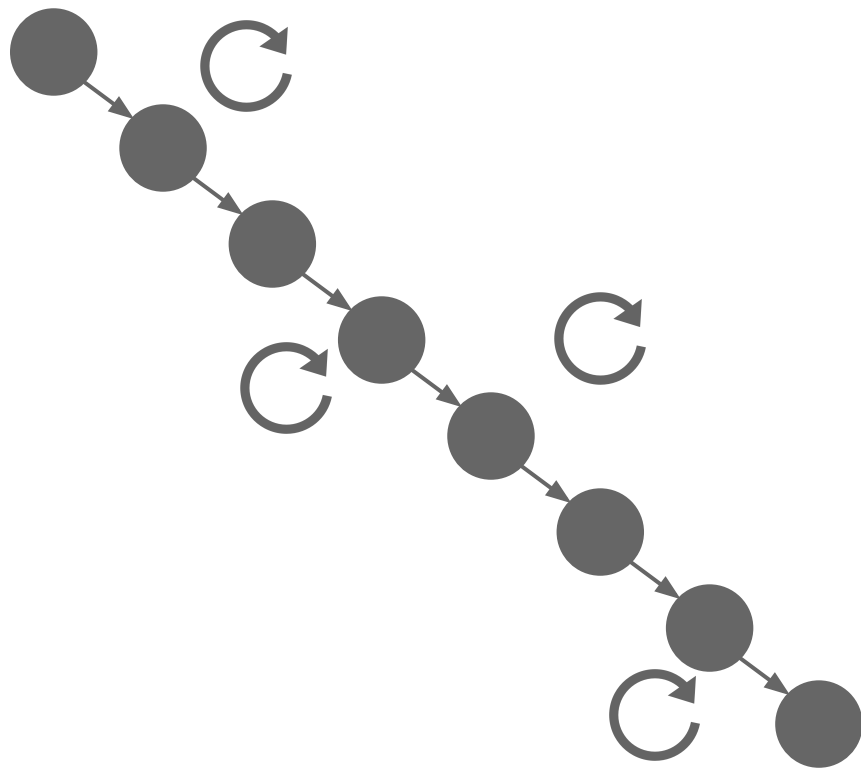
Rotacionar desbalanceada

Aplica rotações

Transforma em backbone

Aplica rotações

Rotaciona até balancear



Rotação

ROTAÇÃO

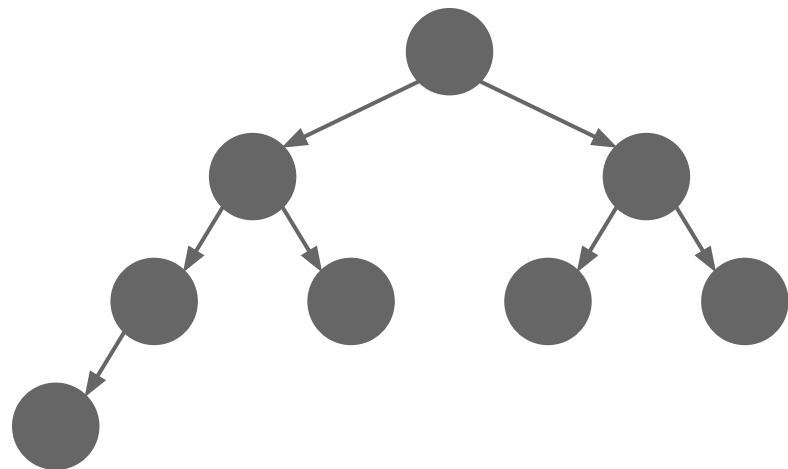
Rotacionar desbalanceada

Aplica rotações

Transforma em backbone

Aplica rotações

Rotaciona até balancear



Rotação

ROTAÇÃO

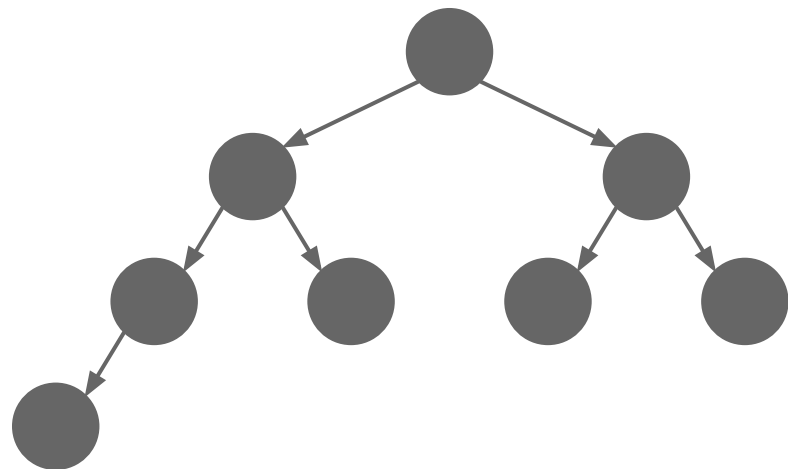
Rotacionar desbalanceada

Aplica rotações

Transforma em backbone

Aplica rotações

Rotaciona até balancear



Rotação

ROTAÇÃO

Rotacionar desbalanceada

Aplica rotações

Transforma em backbone

Aplica rotações

Rotaciona até balancear

Rotação a direita

Rotação a esquerda

Árvore -> backbone

Backbone -> árvore

Rotação

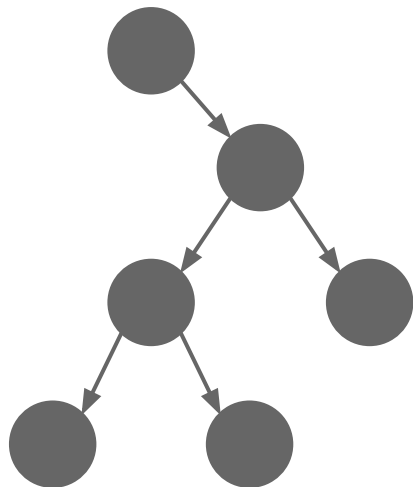
Rotação a direita

Rotação a esquerda

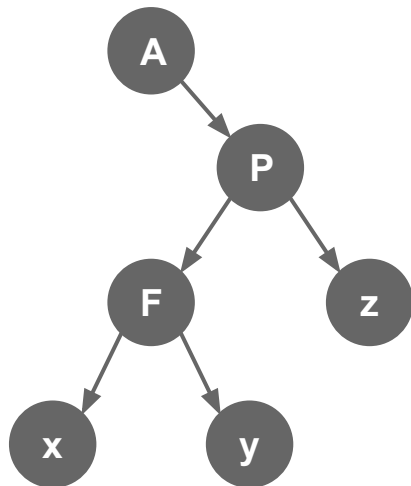
Árvore → backbone

Backbone → árvore

Rotação a direita

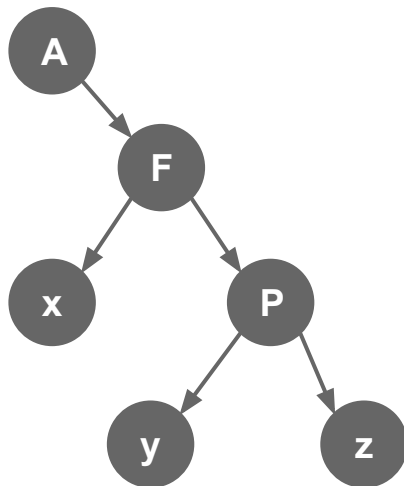


Rotação a direita

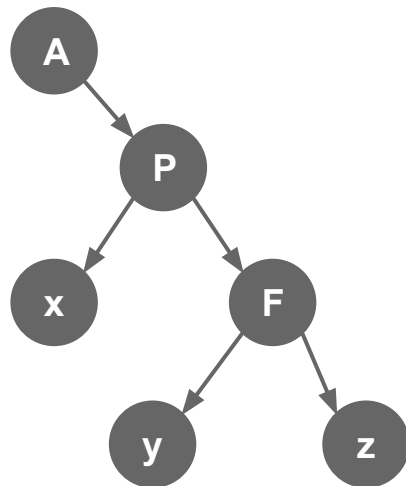


A - avô
P - pai
F - filho

Rotação a direita

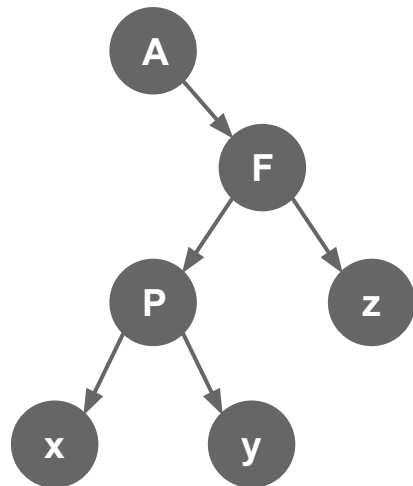


A - avô
P - pai
F - filho



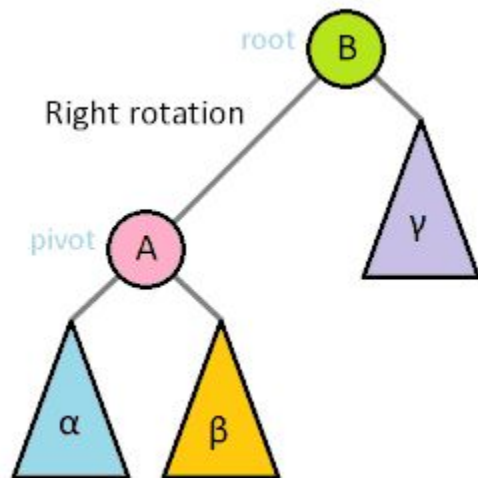
Rotação a esquerda

A - avô
P - pai
F - filho



Rotação a esquerda

A - avô
P - pai
F - filho



Rotação a direita

Rotação a esquerda

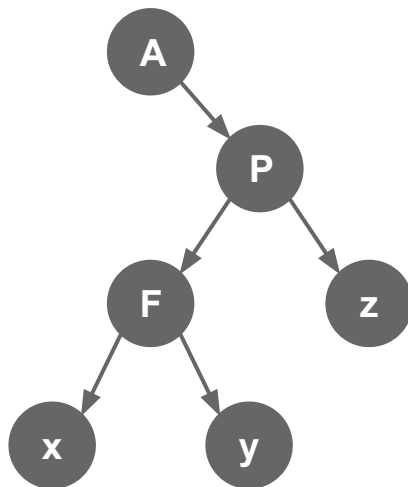
ROTAÇÃO A DIREITA

Rotação a direita

- 1) Recebe: avo, pai e filho
- 2) **SE** avo != NULL:
 - a) **SE** avo.esq == pai
 - i) avo.esq = filho
 - b) **SENÃO**
 - i) avo.dir = filho
- 3) pai.esq = filho.dir
- 4) filho.dir = pai

ROTAÇÃO A DIREITA

- 1) Recebe: avo, pai e filho
- 2) **SE** avo != NULL:
 - a) **SE** avo.esq == pai
 - i) avo.esq = filho
 - b) **SENÃO**
 - i) avo.dir = filho
- 3) pai.esq = filho.dir
- 4) filho.dir = pai

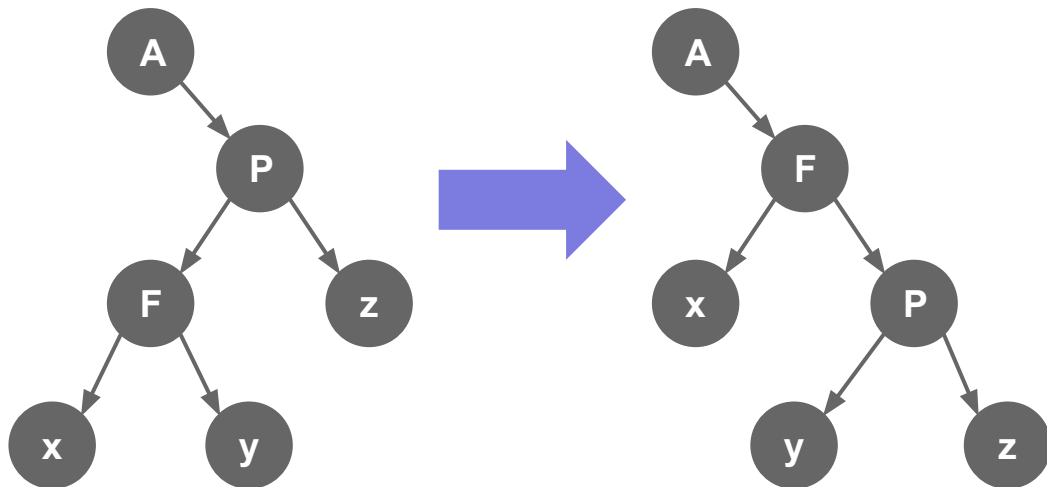


Rotação a direita

ROTAÇÃO A DIREITA

- 1) Recebe: avo, pai e filho
- 2) **SE** avo != NULL:
 - a) **SE** avo.esq == pai
 - i) avo.esq = filho
 - b) **SENÃO**
 - i) avo.dir = filho
- 3) pai.esq = filho.dir
- 4) filho.dir = pai

Rotação a direita



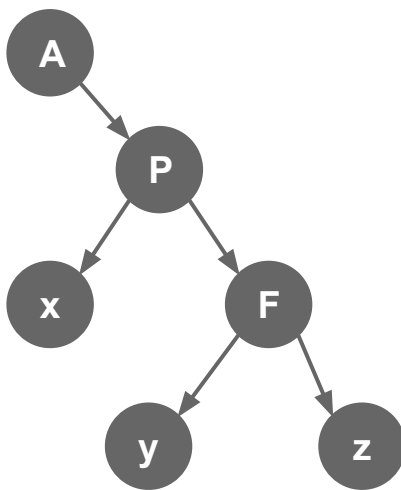
ROTAÇÃO A ESQUERDA

- 1) Recebe: avo, pai e filho
- 2) **SE** avo != NULL:
 - a) **SE** avo.esq == pai
 - i) avo.esq = filho
 - b) **SENÃO**
 - i) avo.dir = filho
- 3) pai.dir = filho.esq
- 4) filho.esq = pai

Rotação a esquerda

ROTAÇÃO A ESQUERDA

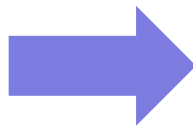
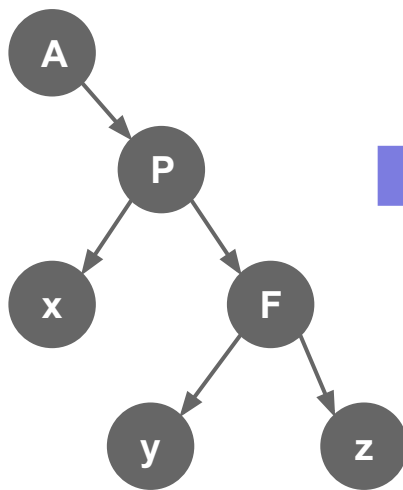
- 1) Recebe: avo, pai e filho
- 2) **SE** avo != NULL:
 - a) **SE** avo.esq == pai
 - i) avo.esq = filho
 - b) **SENÃO**
 - i) avo.dir = filho
- 3) pai.dir = filho.esq
- 4) filho.esq = pai



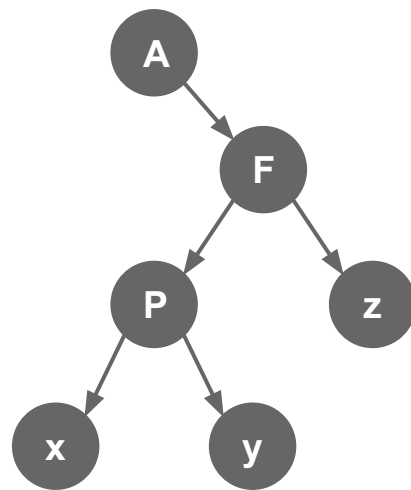
Rotação a esquerda

ROTAÇÃO A ESQUERDA

- 1) Recebe: avo, pai e filho
- 2) **SE** avo != NULL:
 - a) **SE** avo.esq == pai
 - i) avo.esq = filho
 - b) **SENÃO**
 - i) avo.dir = filho
- 3) pai.dir = filho.esq
- 4) filho.esq = pai



Rotação a esquerda





Rotação a direita



Rotação a esquerda



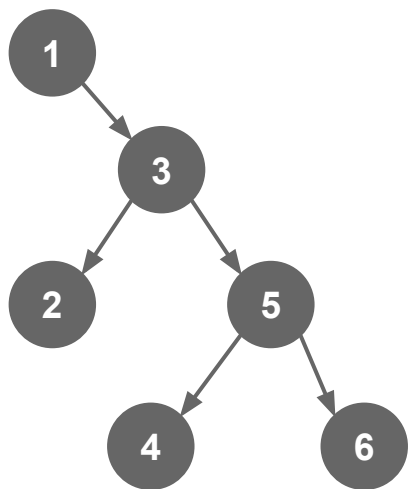


Rotação a direita

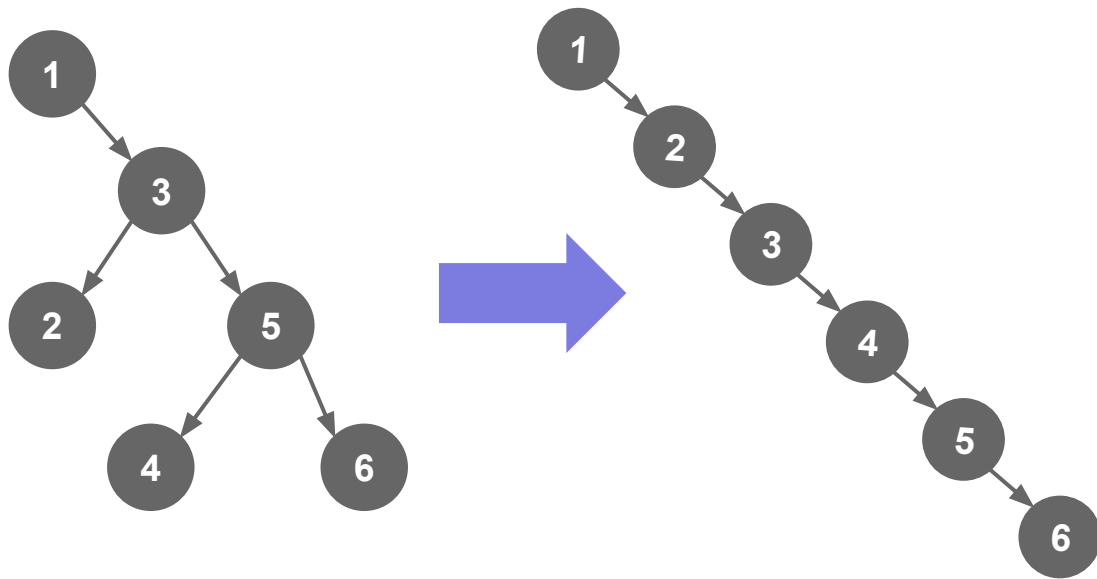


Rotação a esquerda

Árvore -> backbone



Árvore -> backbone

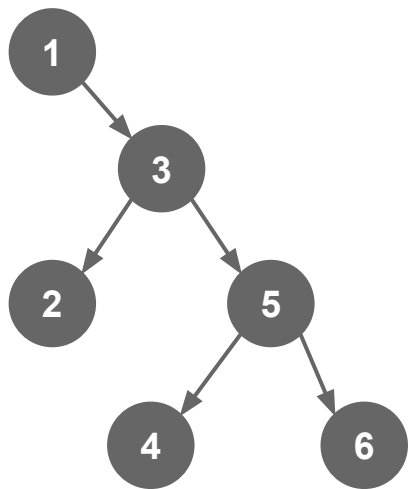


Árvore → backbone

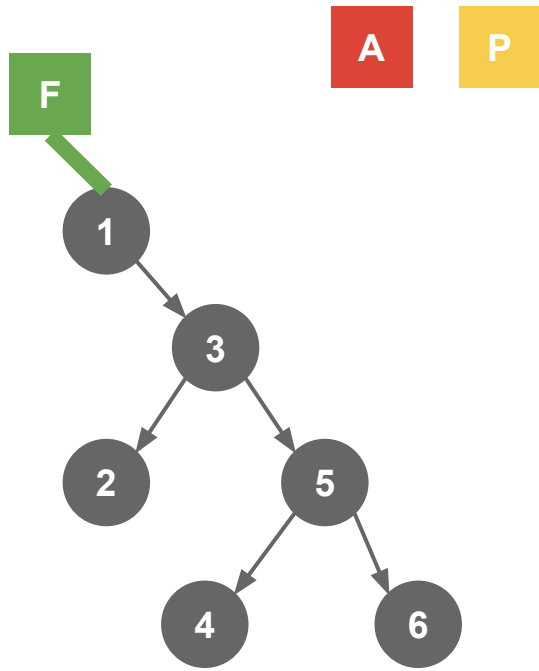
A

P

F



Árvore -> backbone



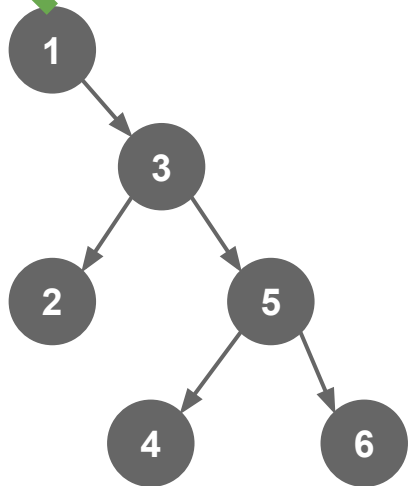
Árvore -> backbone

F

A

P

Tem filho a esquerda?

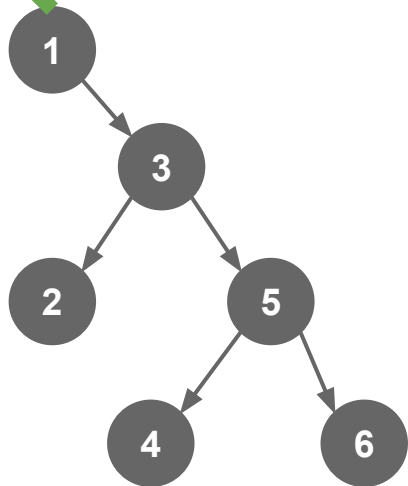


Árvore -> backbone

F

A

P

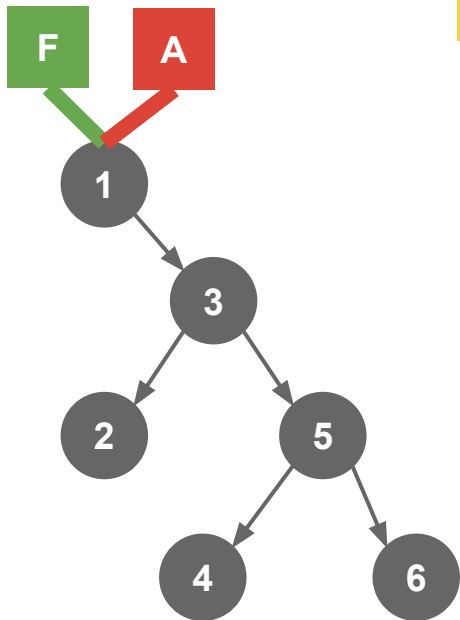


Tem filho a esquerda?

Não

Árvore -> backbone

P



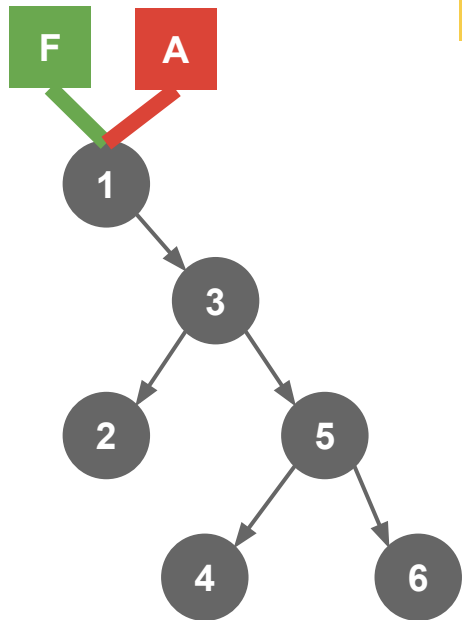
Tem filho a esquerda?

Não

Atualiza o avô

Árvore -> backbone

P



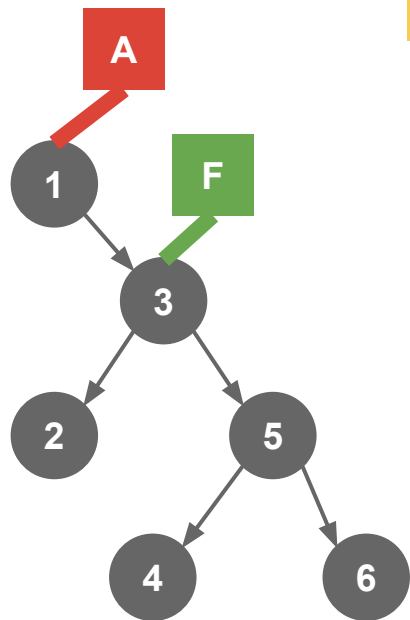
Tem filho a esquerda?

Não

Atualiza o avô

Árvore -> backbone

P



Tem filho a esquerda?

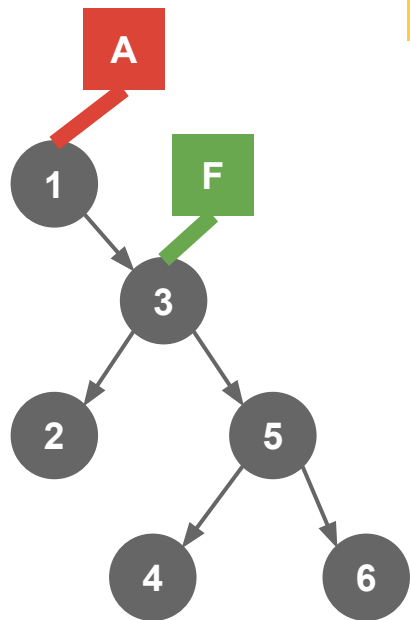
Não

Atualiza o avô

Caminha direita

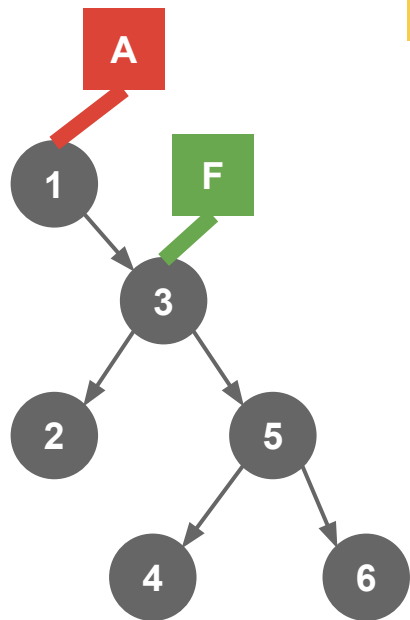
Árvore -> backbone

P



Árvore -> backbone

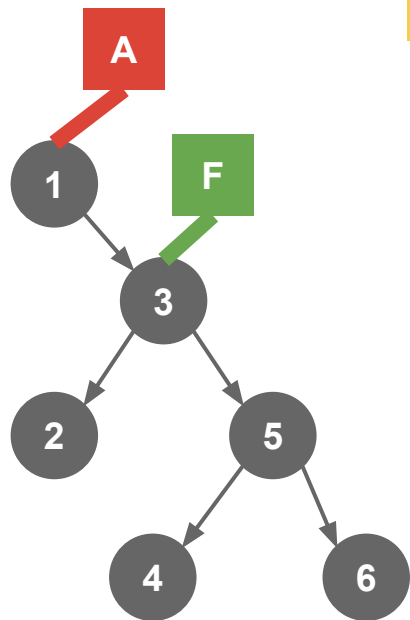
P



Tem filho a esquerda?

Árvore -> backbone

P

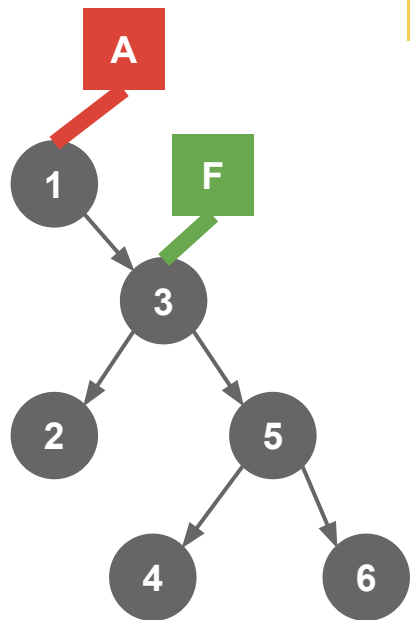


Tem filho a esquerda?

Sim

Árvore -> backbone

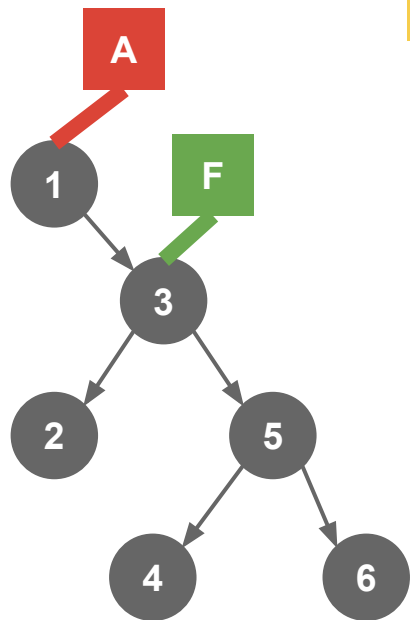
P



É raiz?

Árvore → backbone

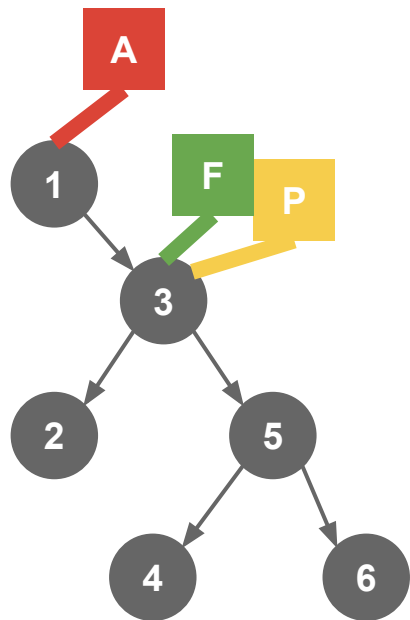
P



É raiz?

Não

Árvore -> backbone

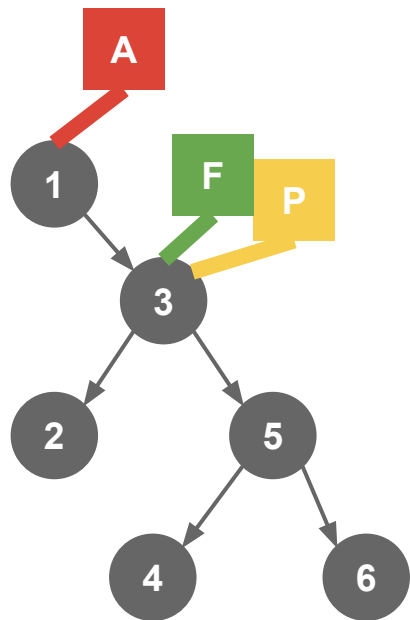


É raiz?

Não

Atualiza o pai

Árvore -> backbone



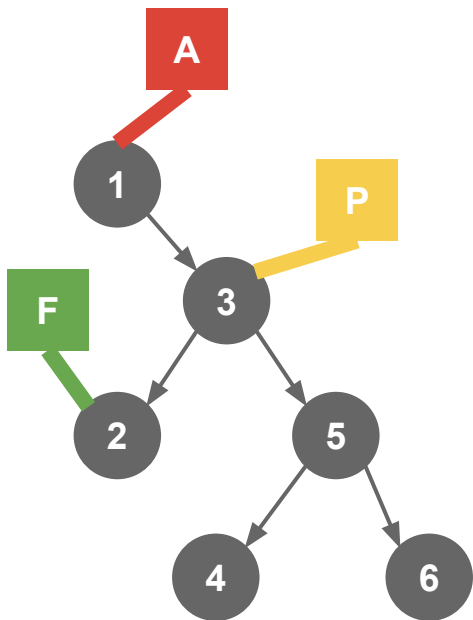
É raiz?

Não

Atualiza o pai

Vai para esquerda

Árvore -> backbone



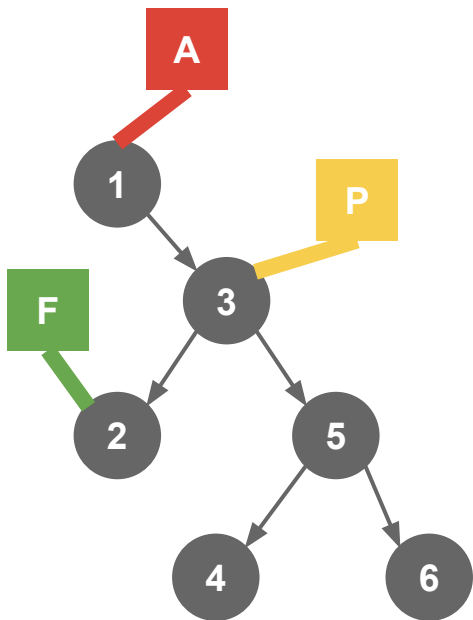
É raiz?

Não

Atualiza o pai

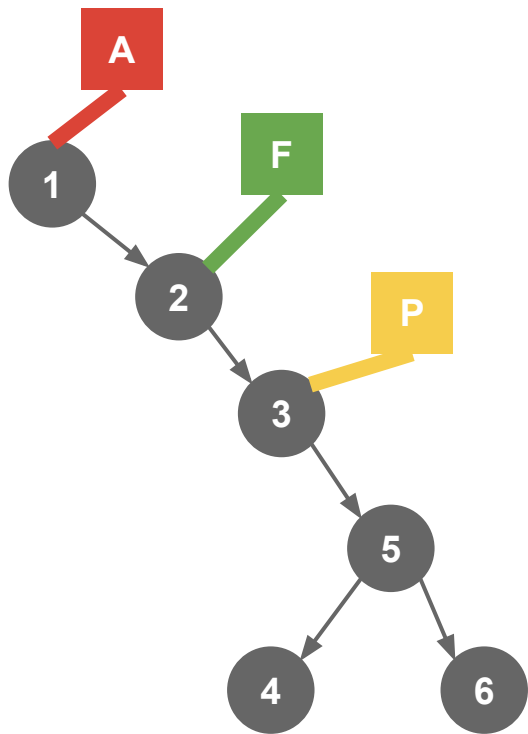
Vai para esquerda

Árvore -> backbone



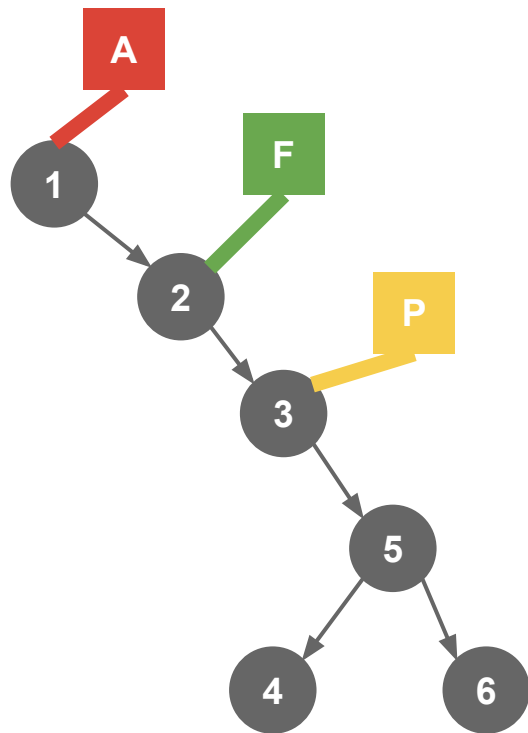
Rotaciona à direita

Árvore -> backbone

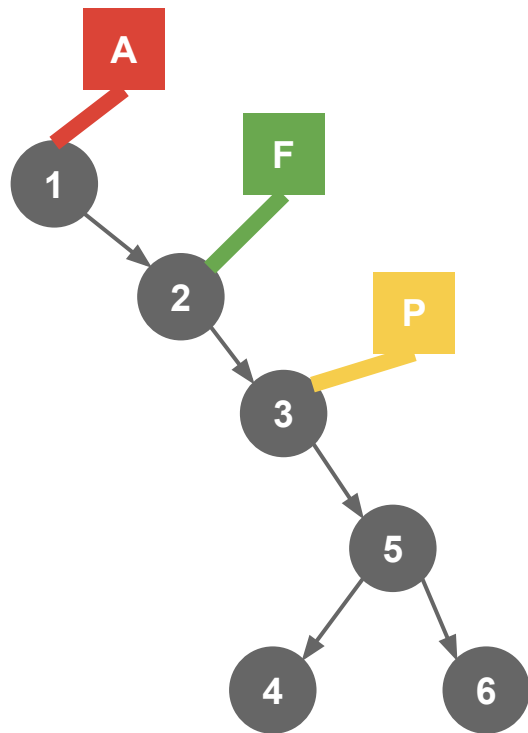


Rotaciona à direita

Árvore -> backbone



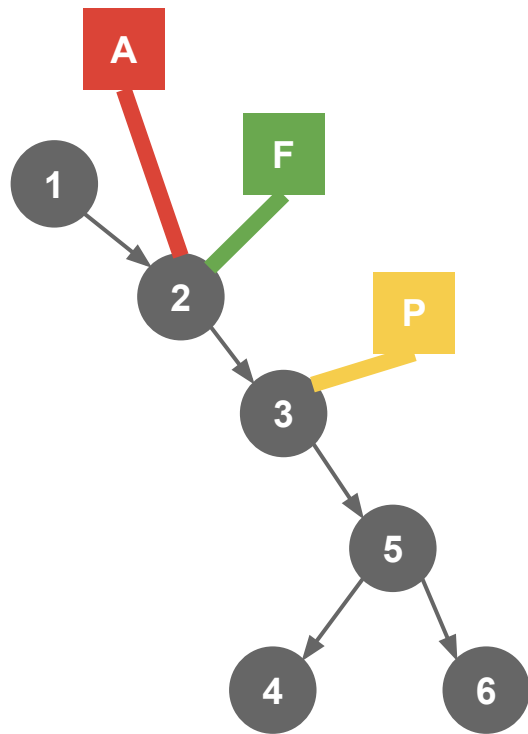
Árvore -> backbone



Tem filho à esquerda?

Não

Árvore -> backbone

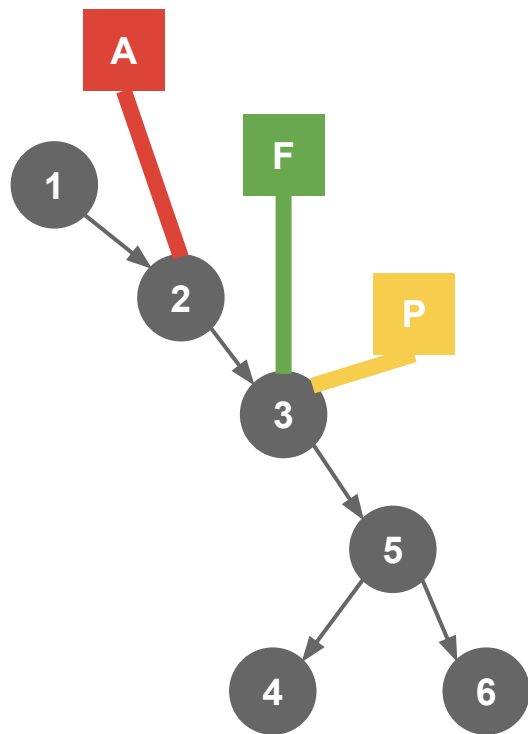


Tem filho à esquerda?

Não

Atualiza o avô

Árvore -> backbone



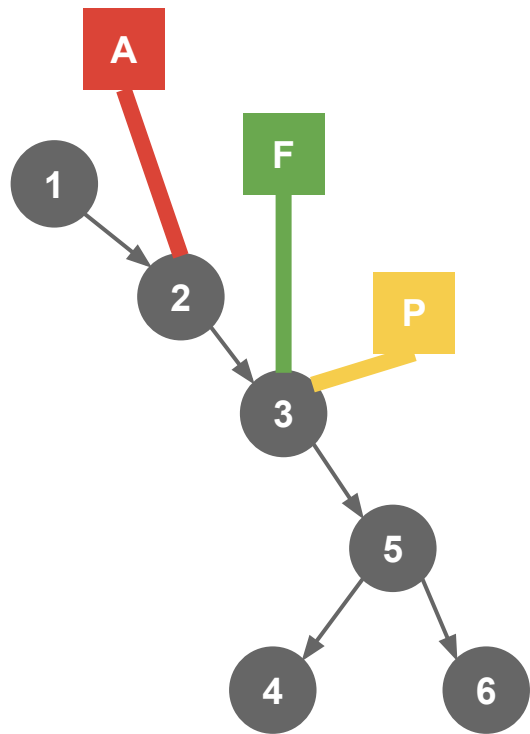
Tem filho à esquerda?

Não

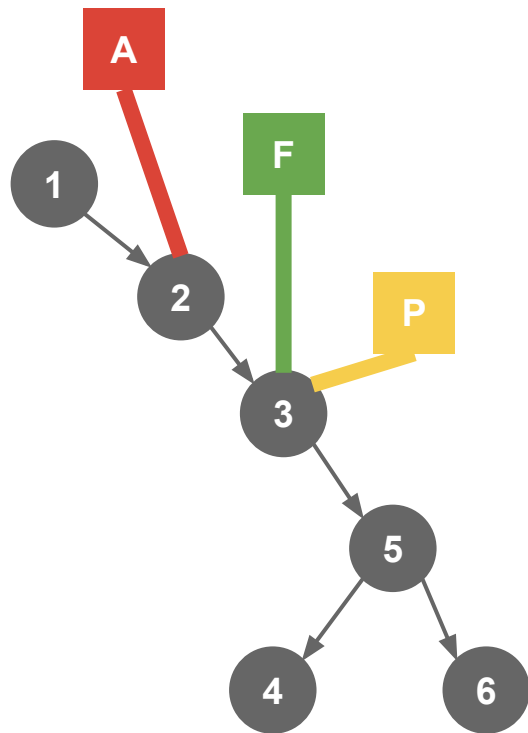
Atualiza o avô

Caminha direita

Árvore -> backbone

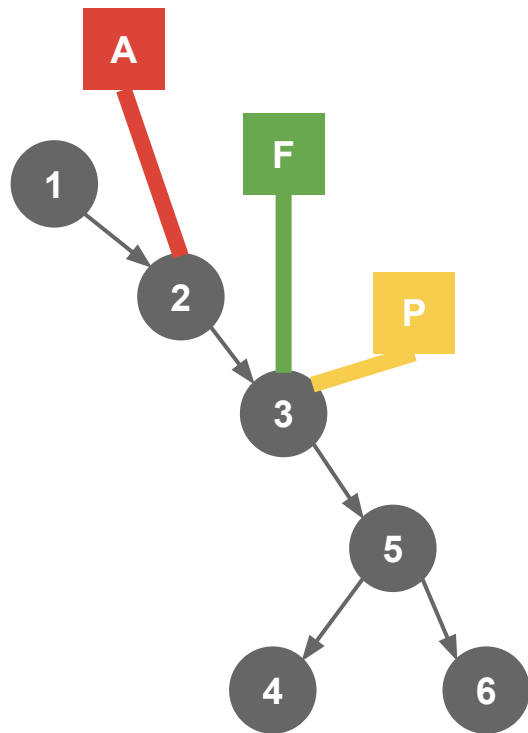


Árvore -> backbone



Tem filho à esquerda?

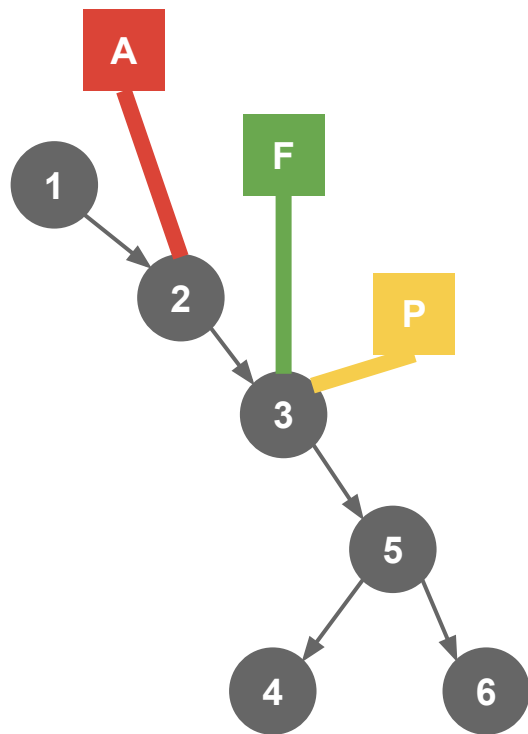
Árvore -> backbone



Tem filho à esquerda?

Não

Árvore -> backbone



Tem filho à esquerda?

Não

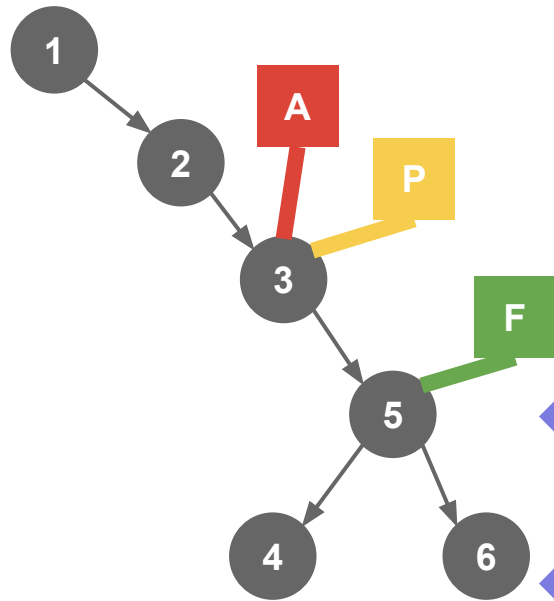
Atualiza o avô

Caminha direita

Árvore -> backbone

Tem filho à esquerda?

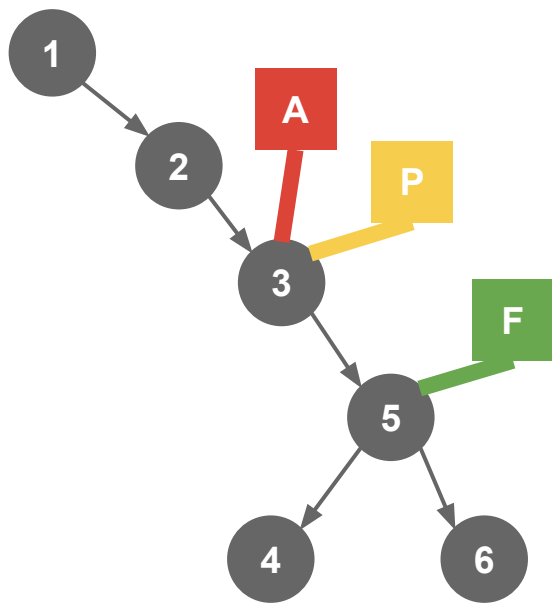
Não



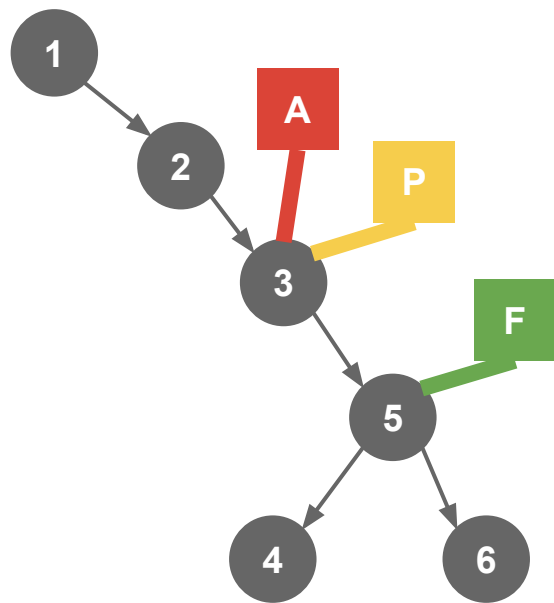
Atualiza o avô

Caminha direita

Árvore -> backbone



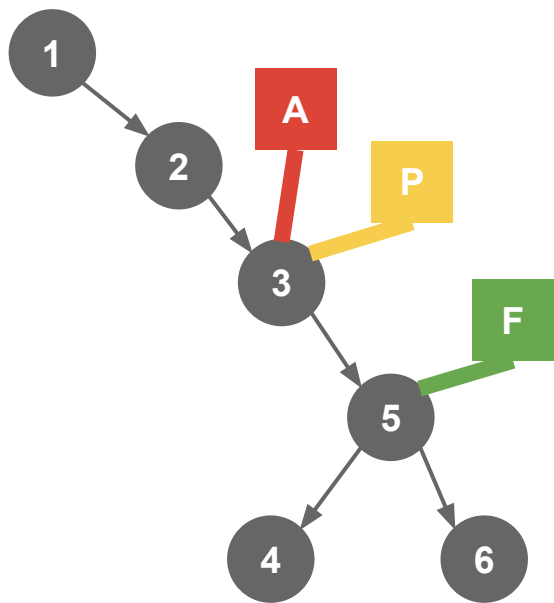
Árvore -> backbone



Tem filho à esquerda?

Sim

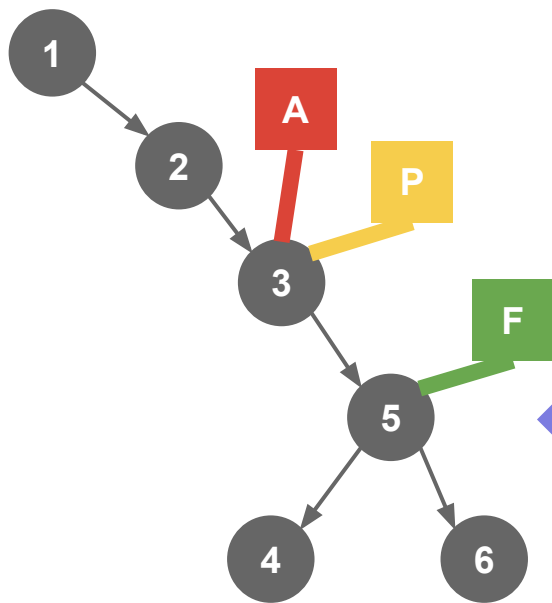
Árvore -> backbone



É raiz?

Não

Árvore → backbone

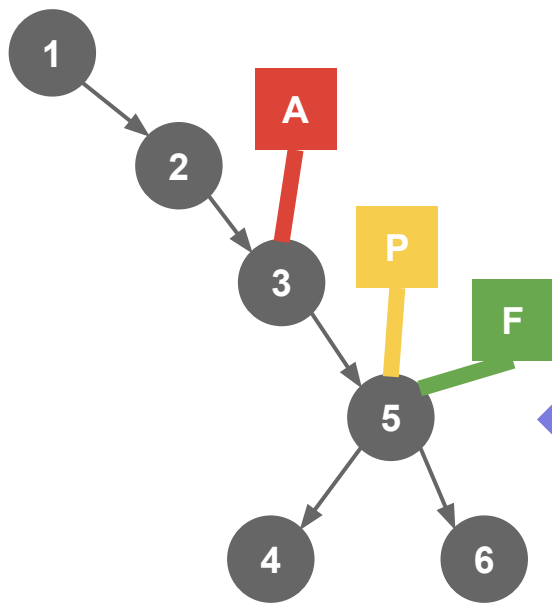


É raiz?

Não

Atualiza o pai

Árvore -> backbone

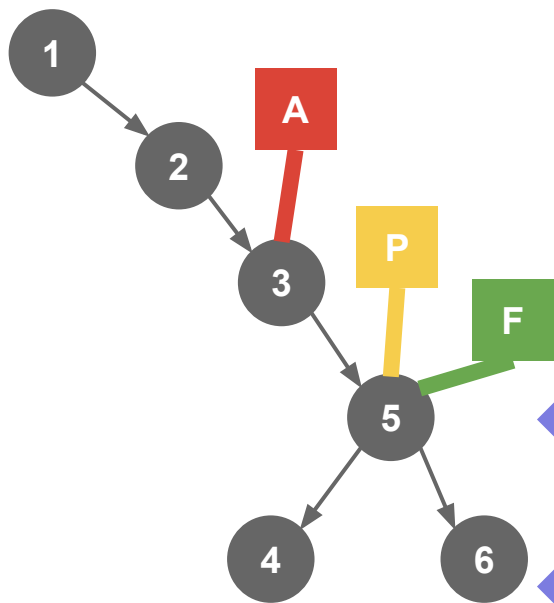


É raiz?

Não

Atualiza o pai

Árvore -> backbone



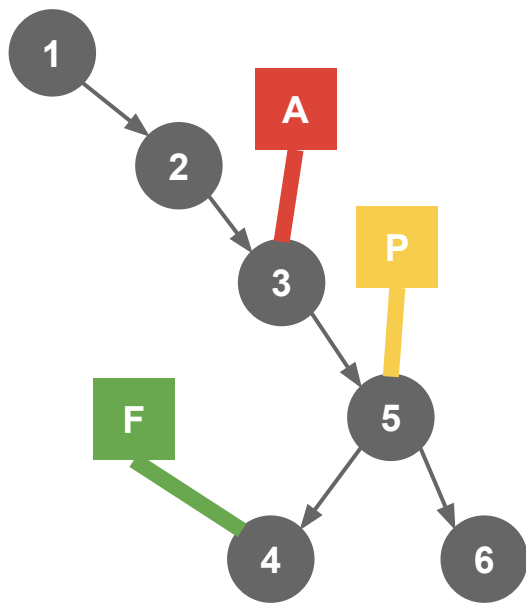
É raiz?

Não

Atualiza o pai

Vai para esquerda

Árvore -> backbone



É raiz?

Não

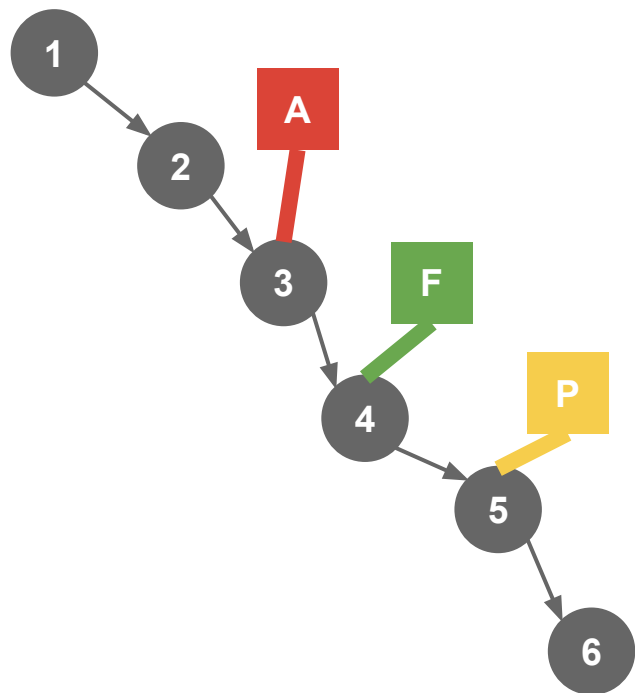
Atualiza o pai

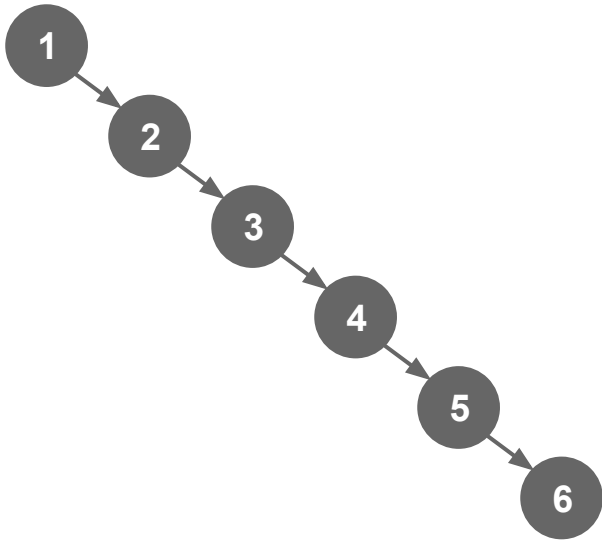
Vai para esquerda

Árvore -> backbone

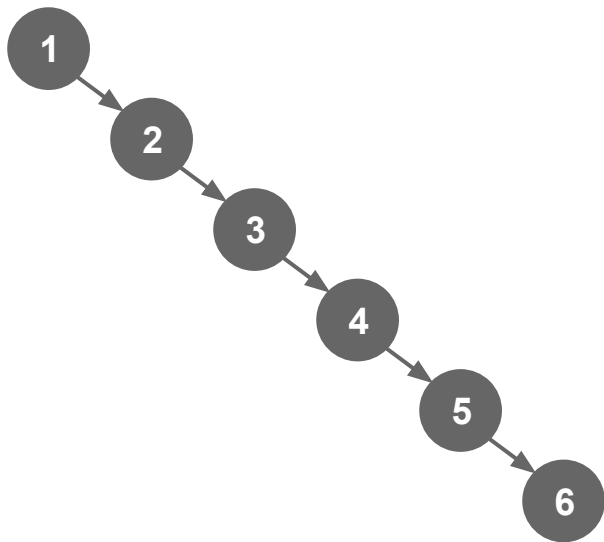
Rotaciona à direita

Árvore -> backbone





Árvore -> backbone

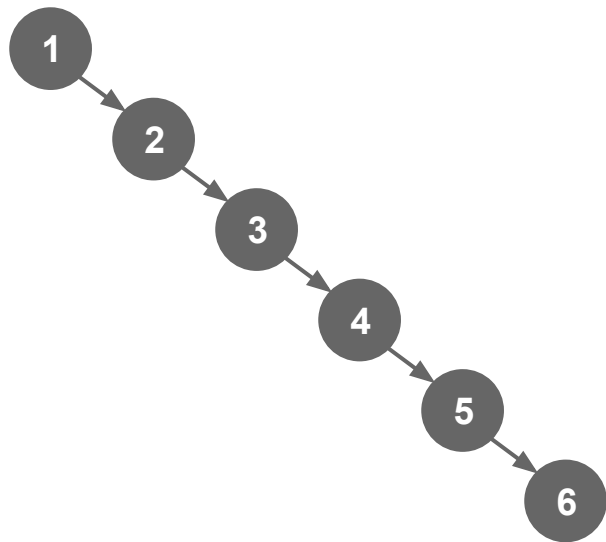


Rotação a direita



Rotação a esquerda

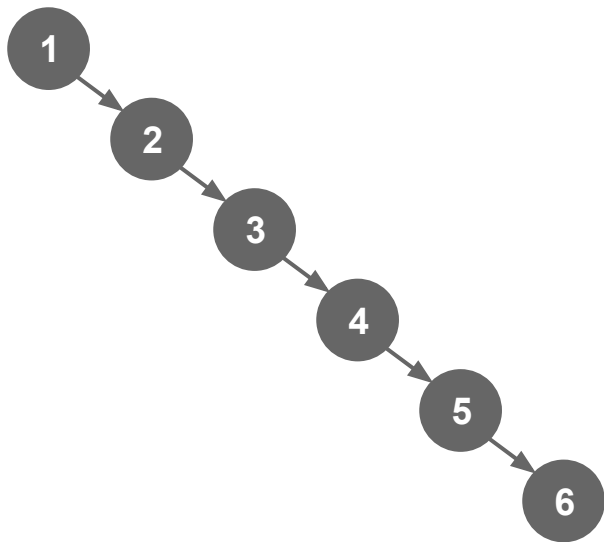
Árvore -> backbone



♥ Rotação a direita

♥ Rotação a esquerda

♥ Árvore → backbone



Rotação a direita

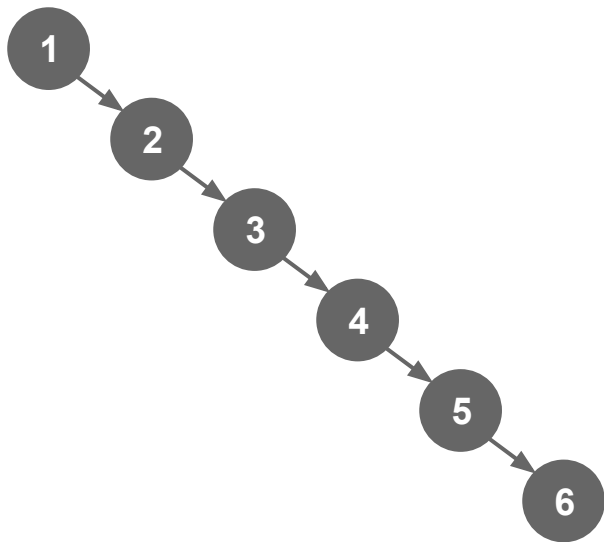


Rotação a esquerda



Árvore -> backbone





Rotação a direita

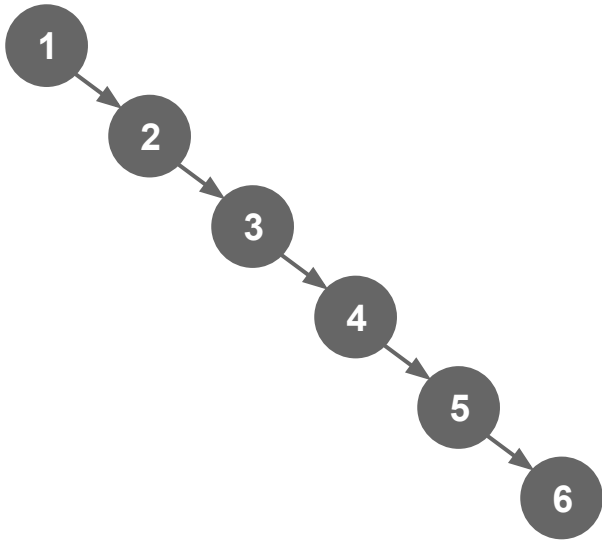


Rotação a esquerda



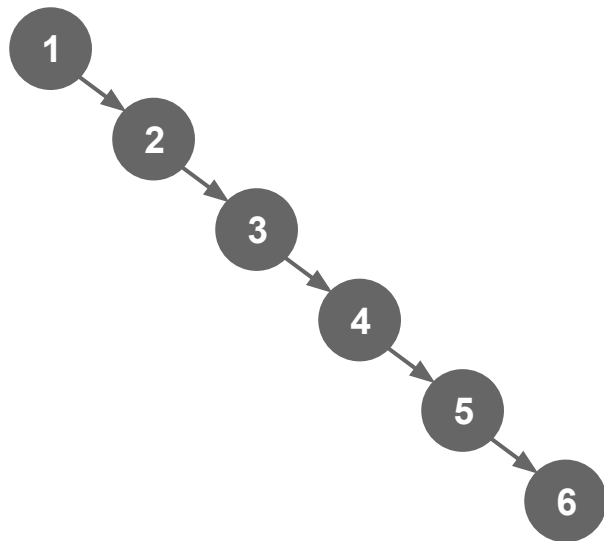
Árvore -> backbone

Backbone -> árvore



Backbone -> árvore

Transforma (3)

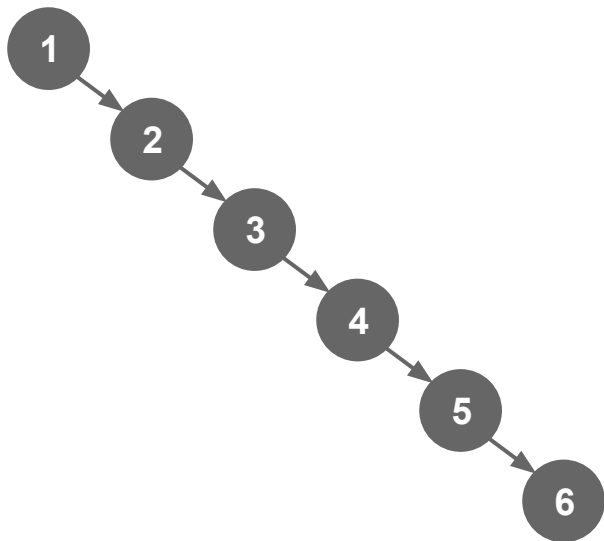


Backbone -> árvore

A

P

F

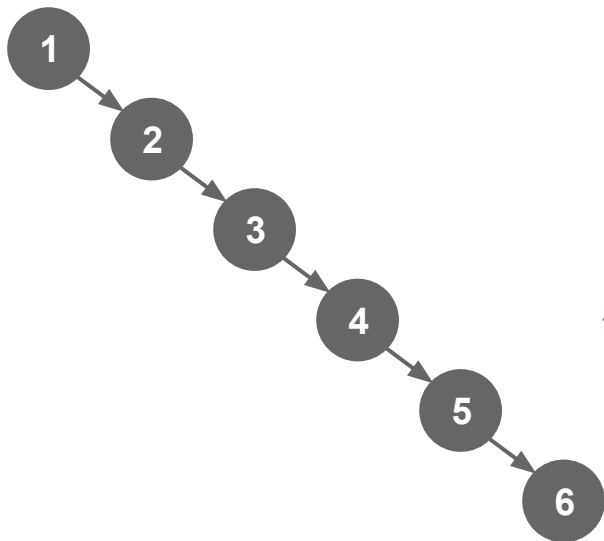


Backbone -> árvore

A

P

F



Começa na raiz

Backbone -> árvore

A

P

F

1

2

3

4

5

6

Começa na raiz

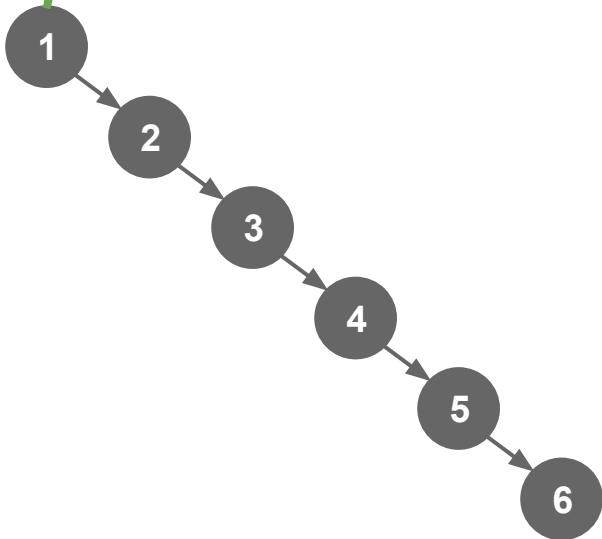
Backbone -> árvore

1

A

P

F



Backbone -> árvore

A

F

P

1

2

3

4

5

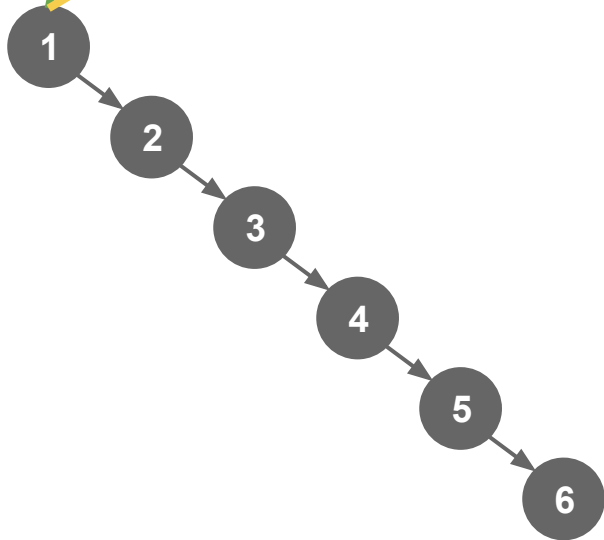
6

Atualiza o pai

Backbone -> árvore

1

A



Atualiza o pai

Vai para direita

Backbone -> árvore

A

P

F

1

2

3

4

5

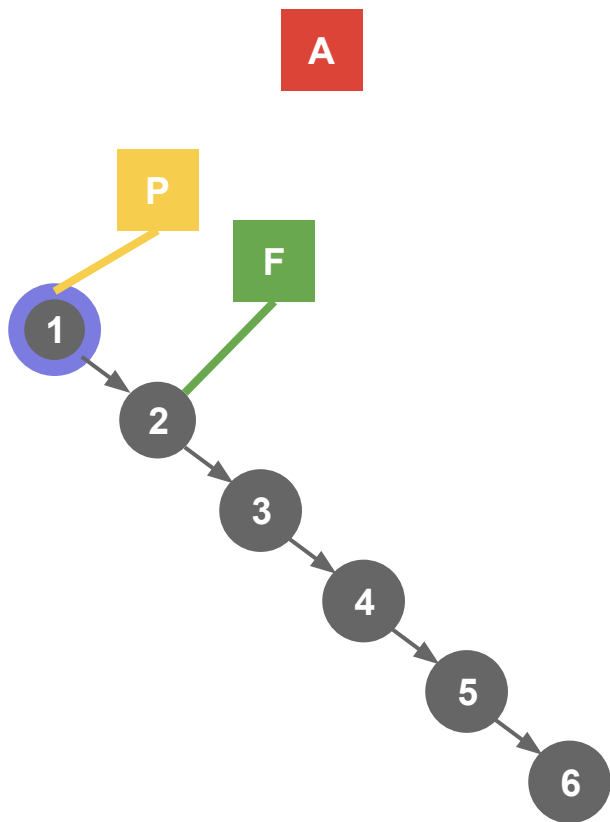
6

Atualiza o pai

Vai para direita

Backbone -> árvore

1



Pai é raiz?

Sim

Backbone -> árvore

A

P

F

Pai é raiz?

Sim

Atualiza raiz

Backbone -> árvore

A

P

F

Pai é raiz?

Sim

Atualiza raiz

Backbone -> árvore

A

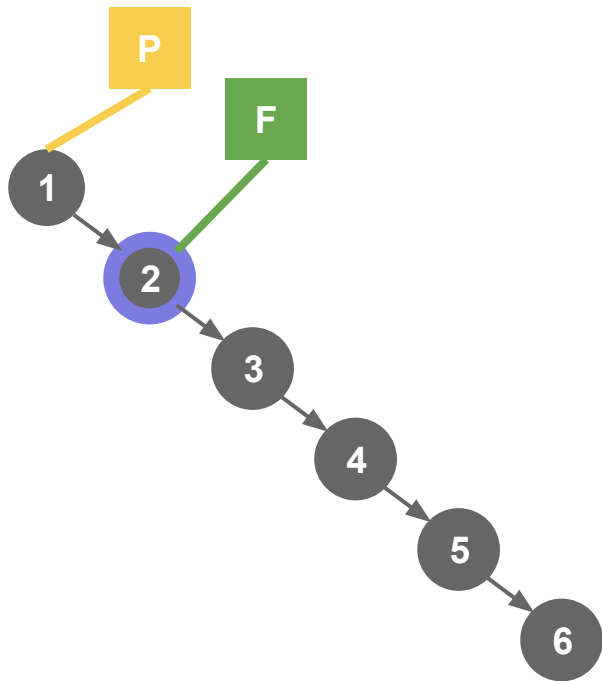
P

F

Rotaciona esquerda

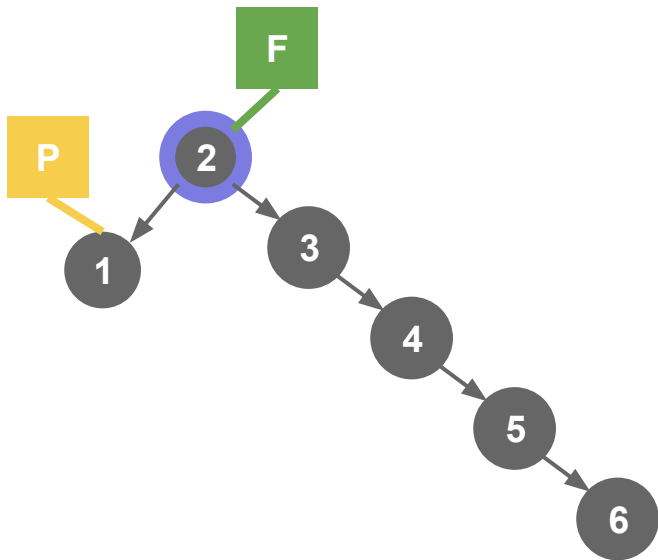
Backbone -> árvore

1



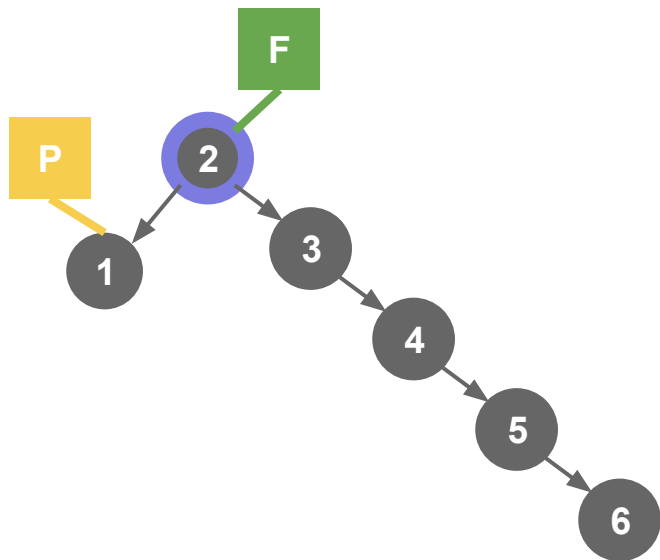
A

Rotaciona esquerda



Backbone -> árvore

A



Transforma

Backbone -> árvore

A

P

F

1

2

3

4

5

6

Backbone -> árvore

A

P

F

2

1

3

4

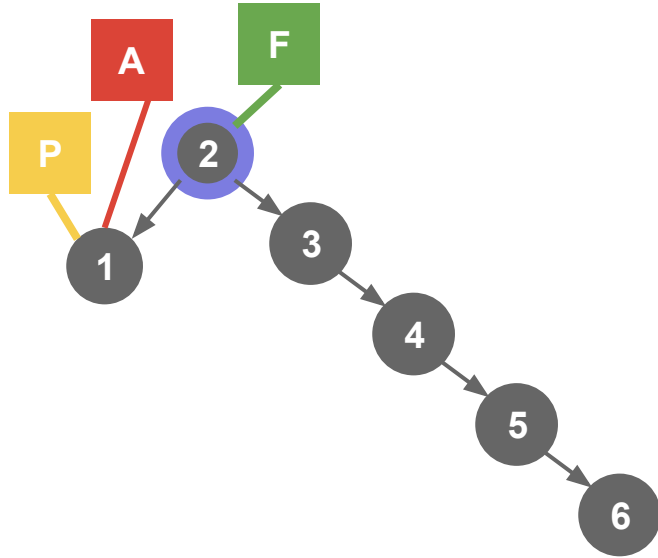
5

6

Atualiza avô

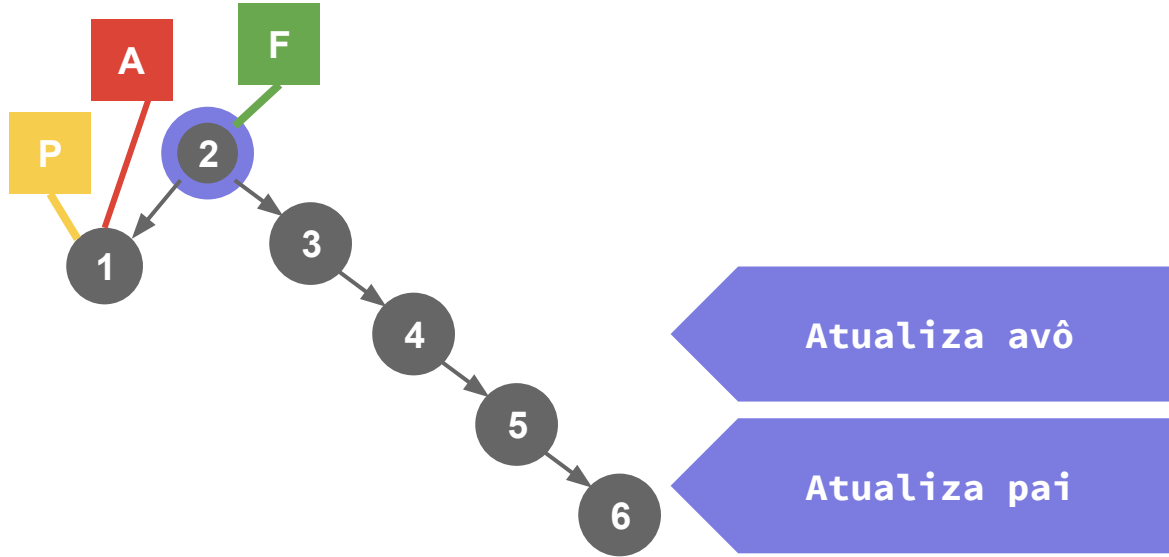
Backbone -> árvore

2

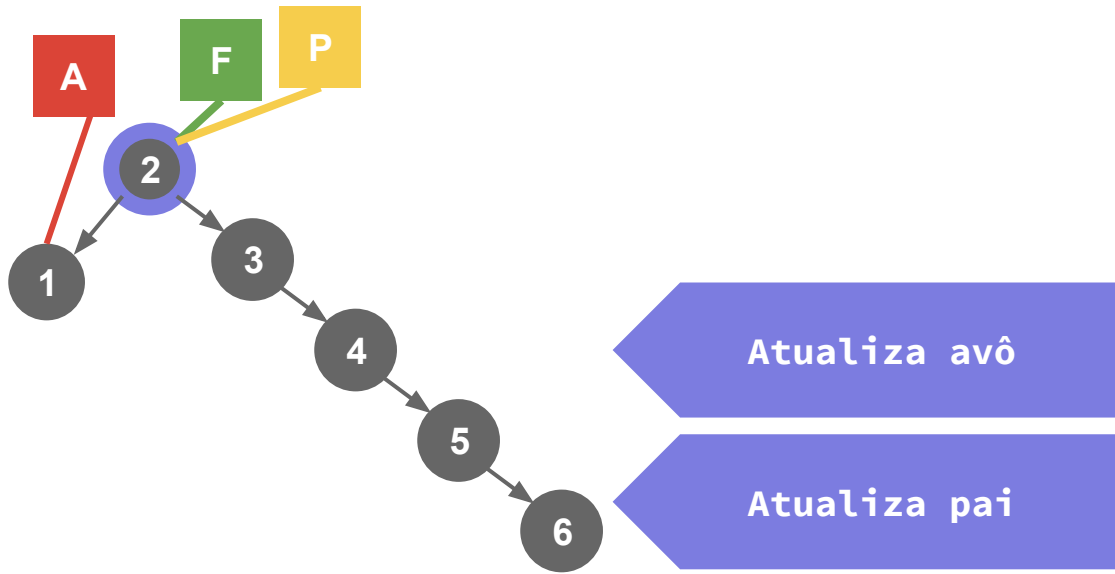


Atualiza avô

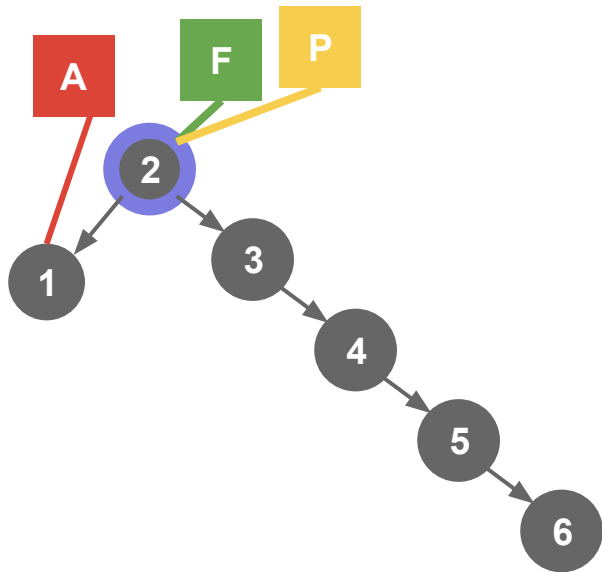
Backbone -> árvore



Backbone -> árvore



Backbone -> árvore

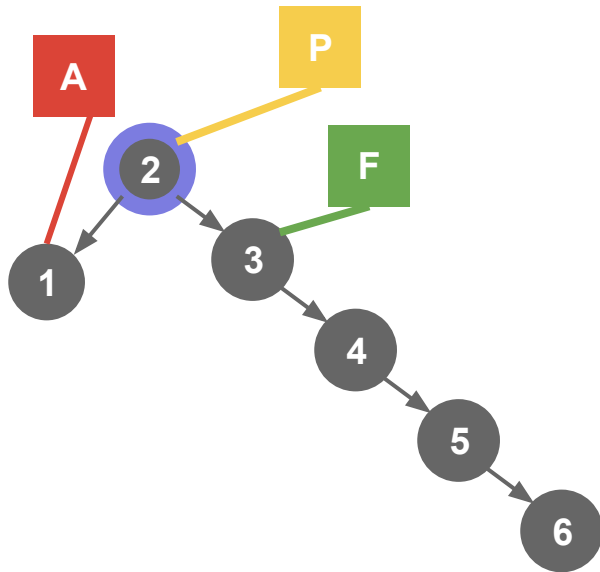


Atualiza avô

Atualiza pai

Vai para direita

Backbone -> árvore

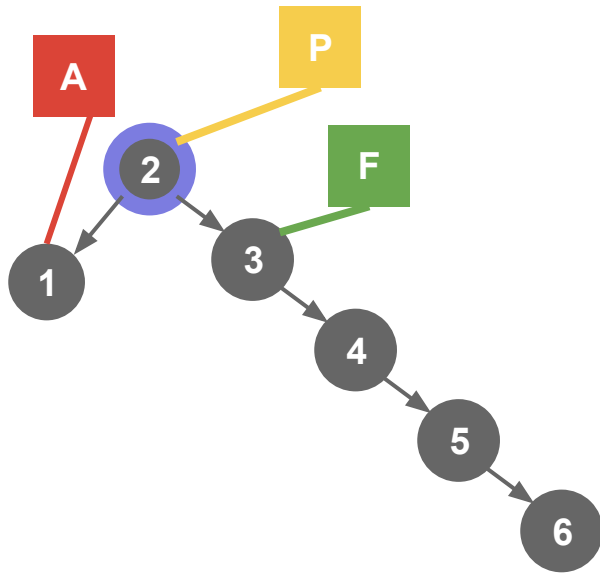


Atualiza avô

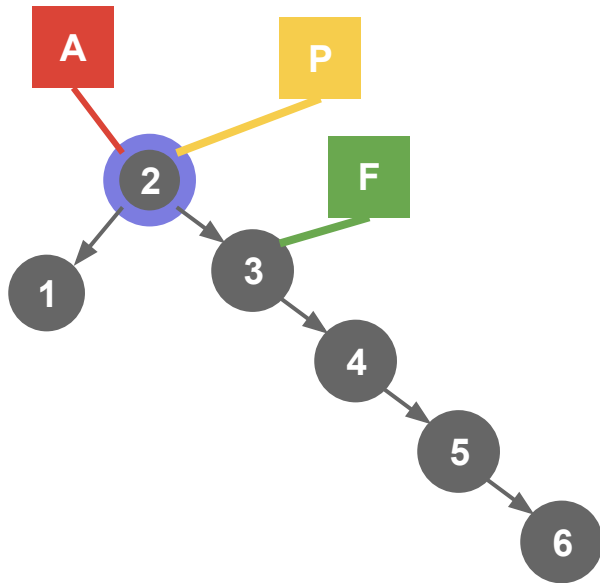
Atualiza pai

Vai para direita

Backbone -> árvore

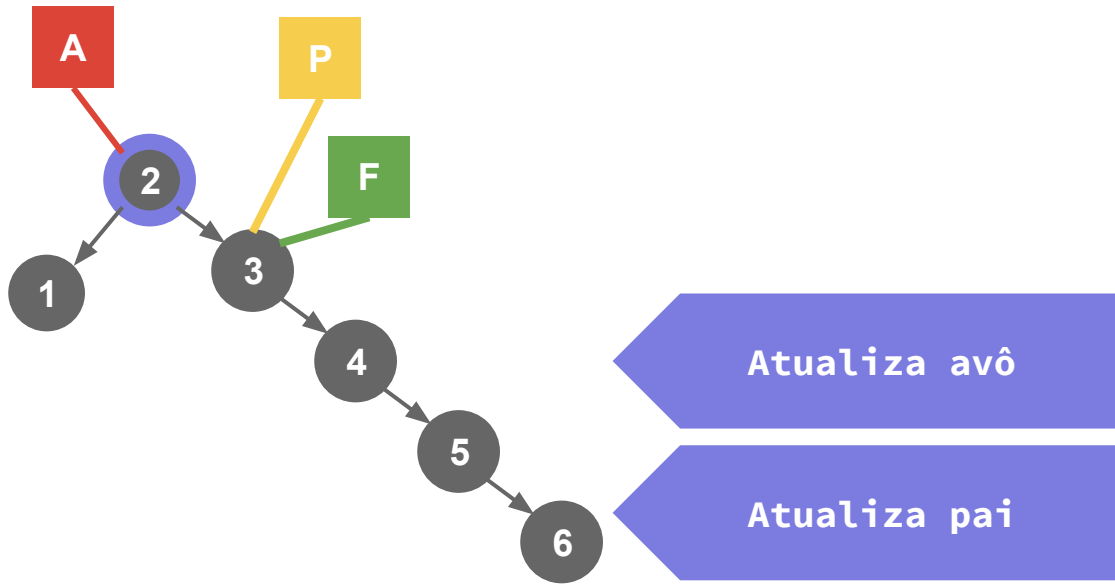


Backbone -> árvore

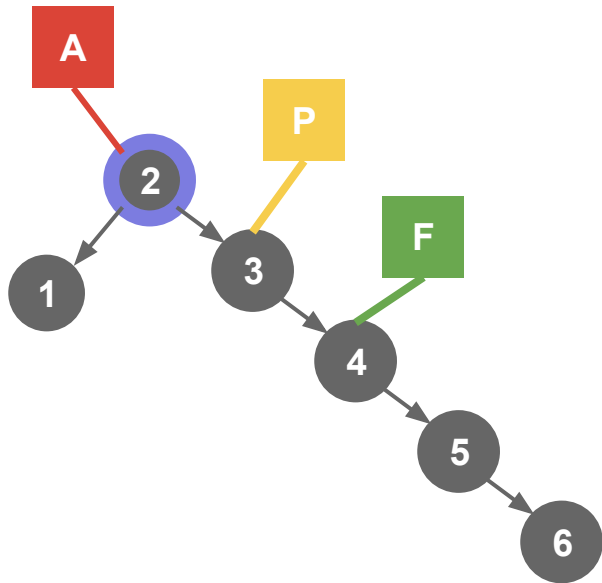


Atualiza avô

Backbone -> árvore



Backbone -> árvore

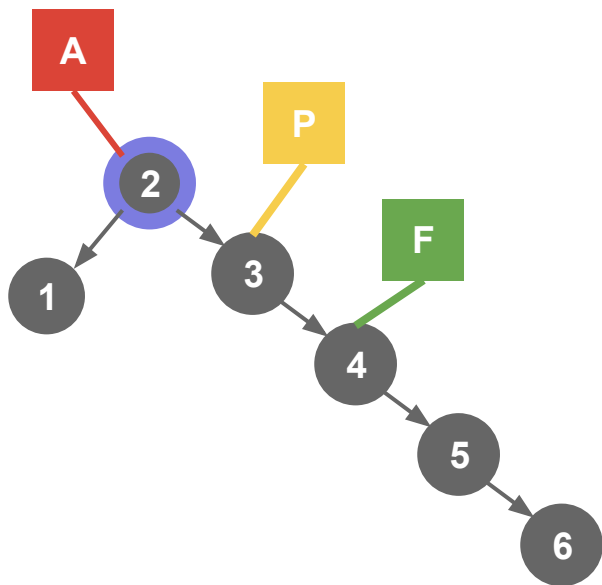


Atualiza avô

Atualiza pai

Vai para direita

Backbone -> árvore

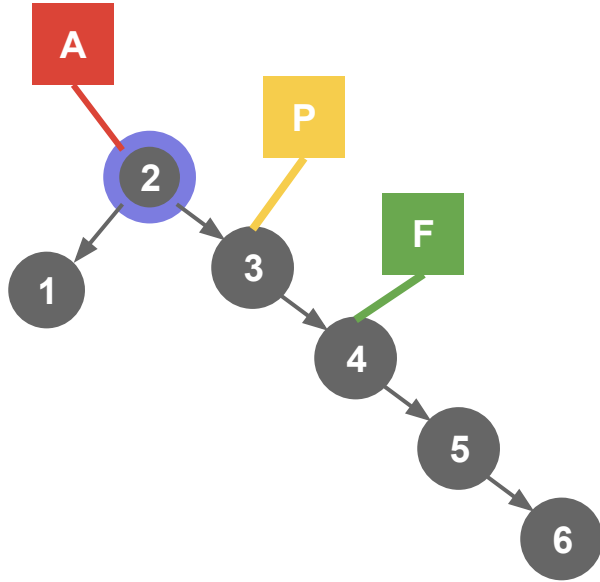


Pai é raiz?

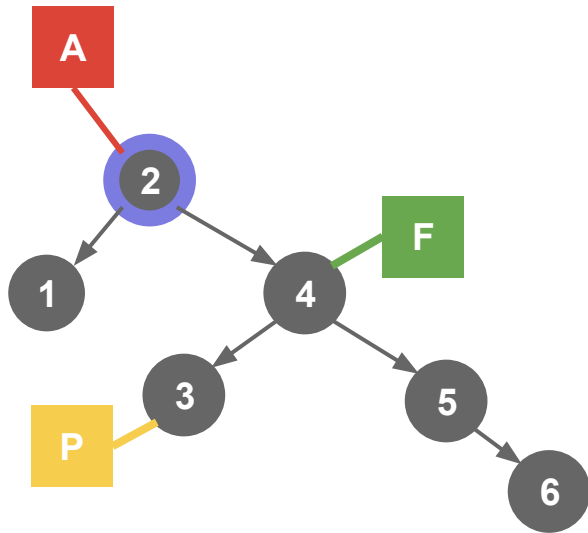
Não

Backbone -> árvore

Rotaciona esquerda

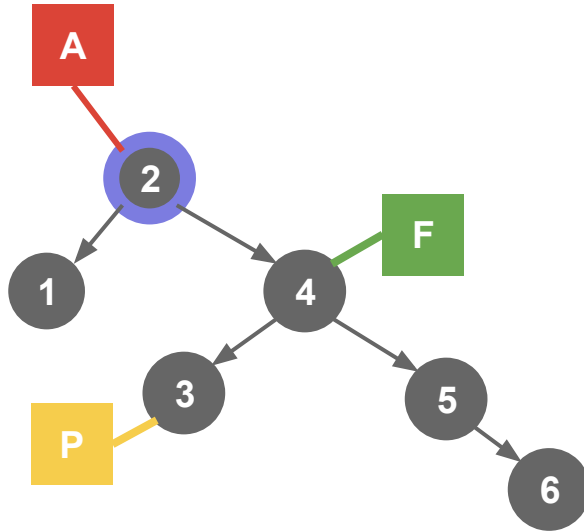


Backbone -> árvore

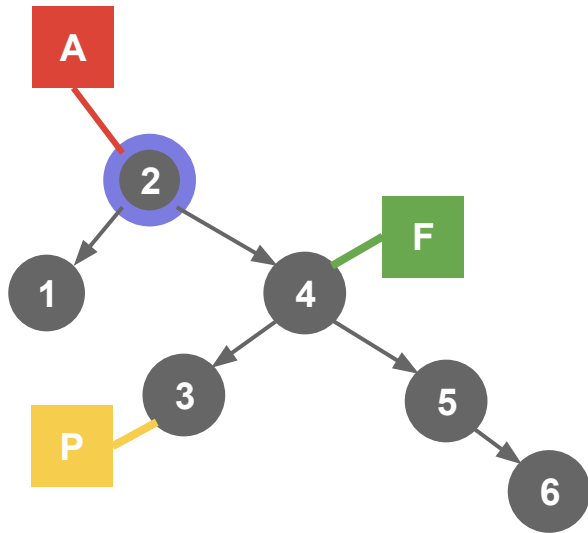


Backbone -> árvore

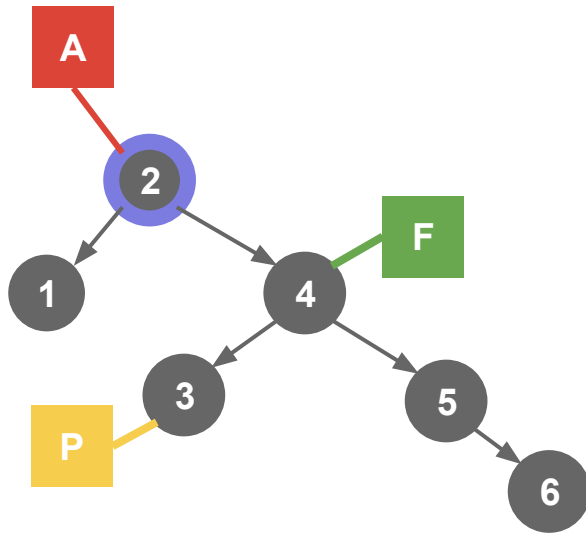
Transforma



Backbone -> árvore

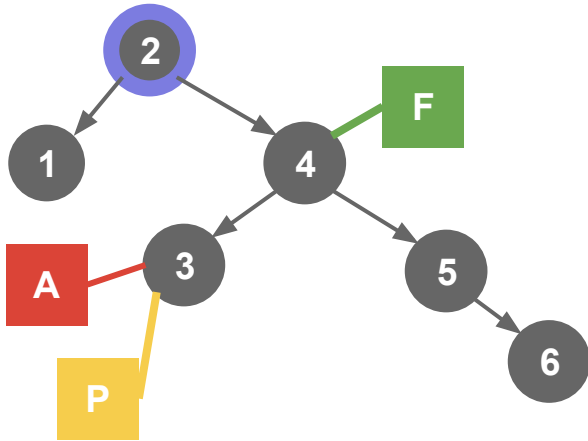


Backbone -> árvore



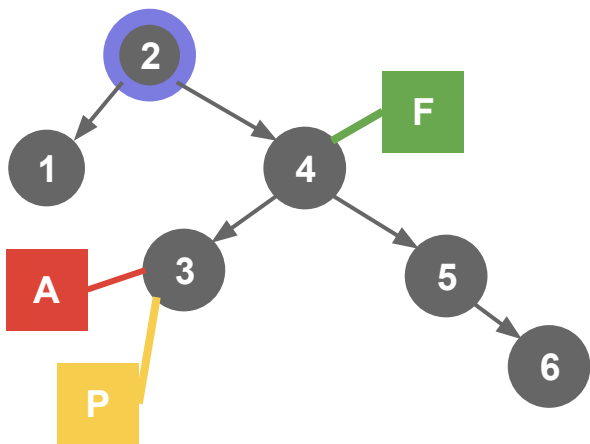
Atualiza avô

Backbone -> árvore



Atualiza avô

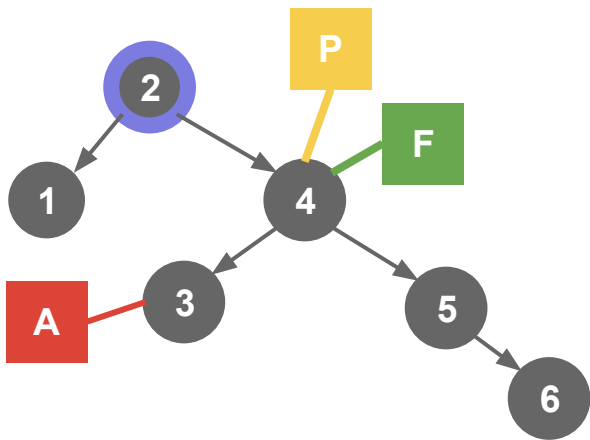
Backbone -> árvore



Atualiza avô

Atualiza pai

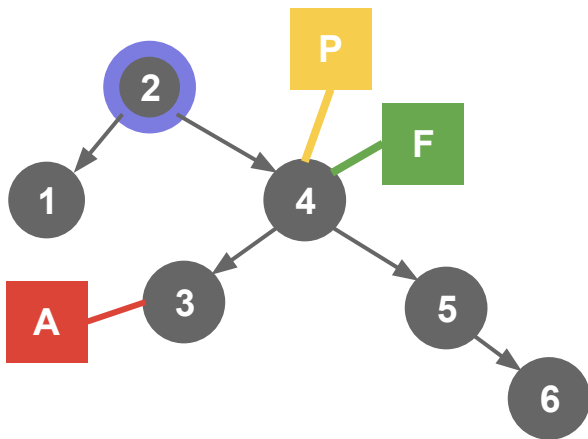
Backbone -> árvore



Atualiza avô

Atualiza pai

Backbone -> árvore

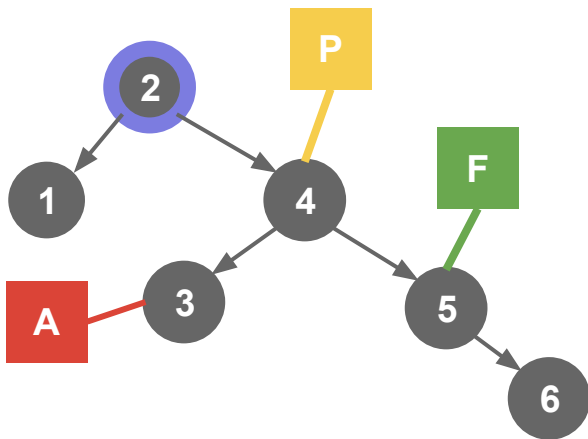


Atualiza avô

Atualiza pai

Vai para direita

Backbone -> árvore

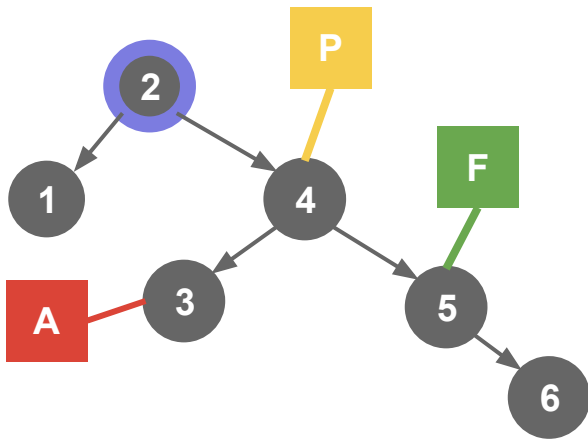


Atualiza avô

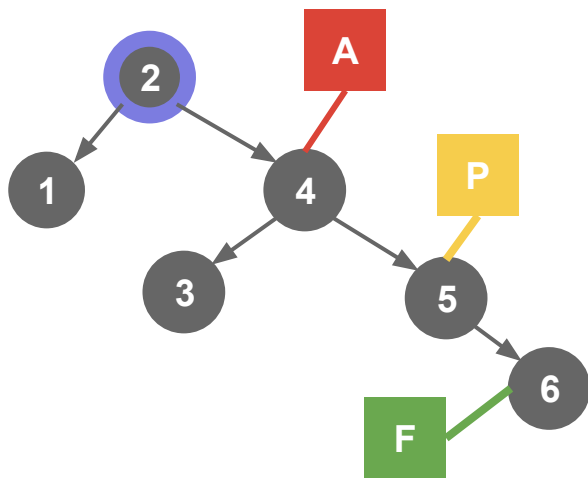
Atualiza pai

Vai para direita

Backbone -> árvore



Backbone -> árvore

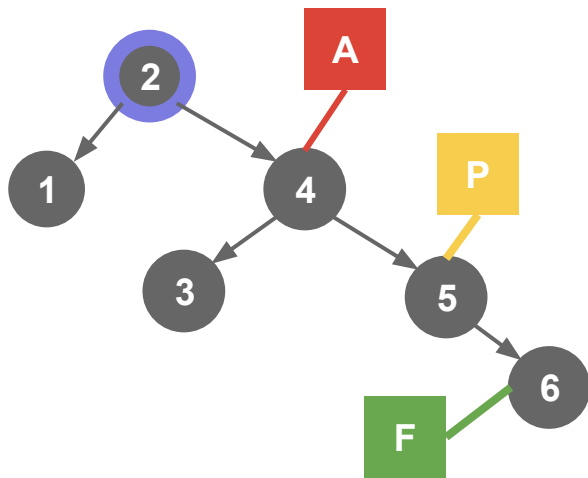


Atualiza avô

Atualiza pai

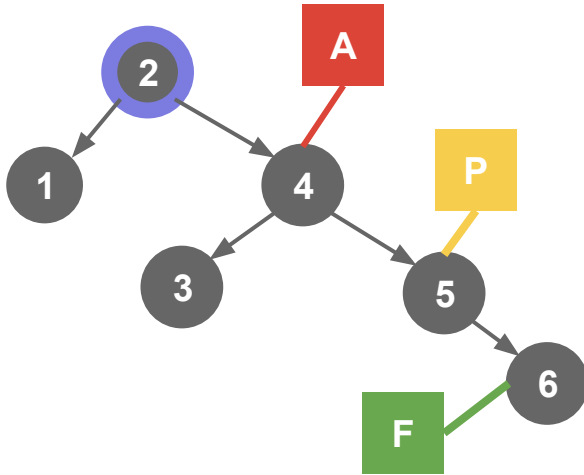
Vai para direita

Backbone -> árvore



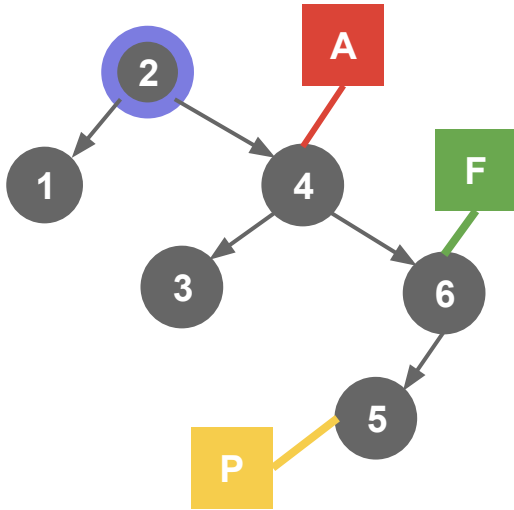
Backbone -> árvore

Rotaciona esquerda

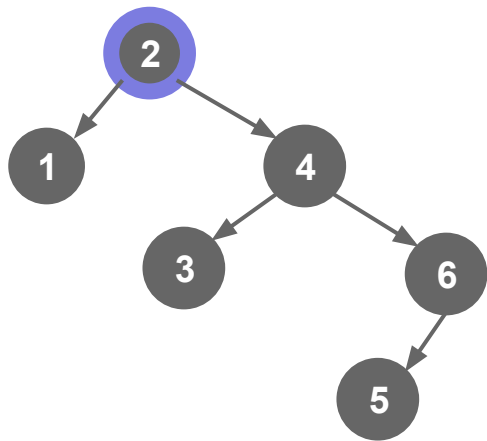


Backbone -> árvore

Rotaciona esquerda



Backbone -> árvore



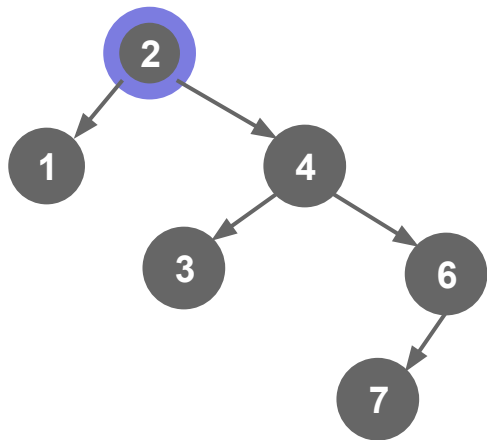
Backbone -> árvore

A

P

F

Transforma(1)

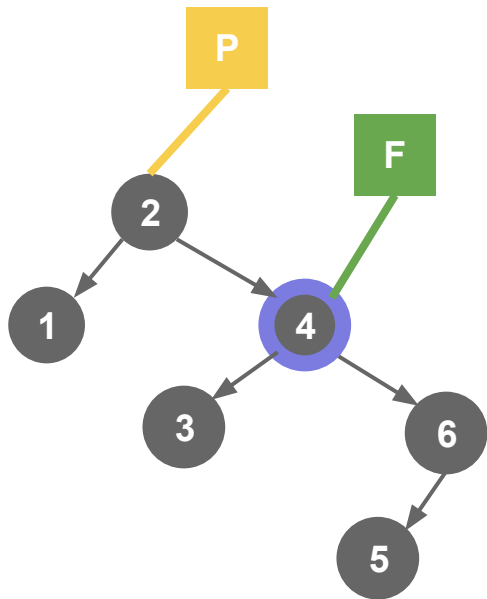


Backbone -> árvore

A

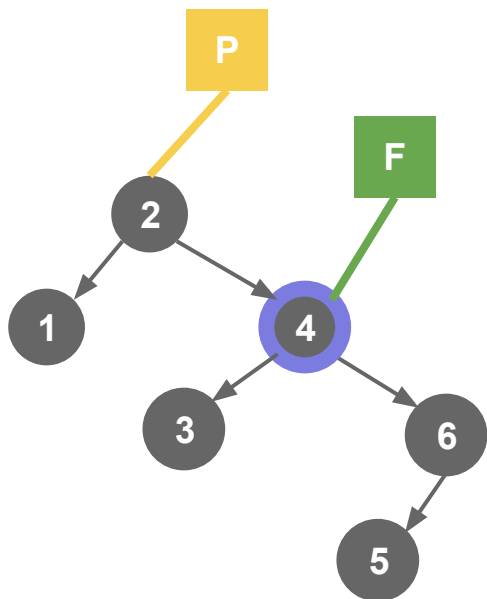
P

F



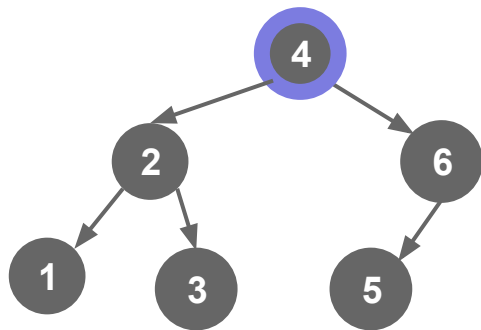
Backbone -> árvore

A

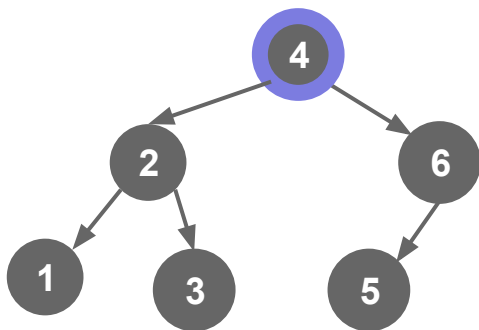


Rotaciona esquerda

Backbone -> árvore



Backbone -> árvore



Rotação a direita

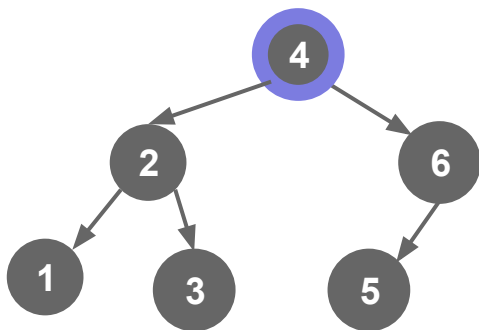


Rotação a esquerda



Árvore -> backbone

Backbone -> árvore

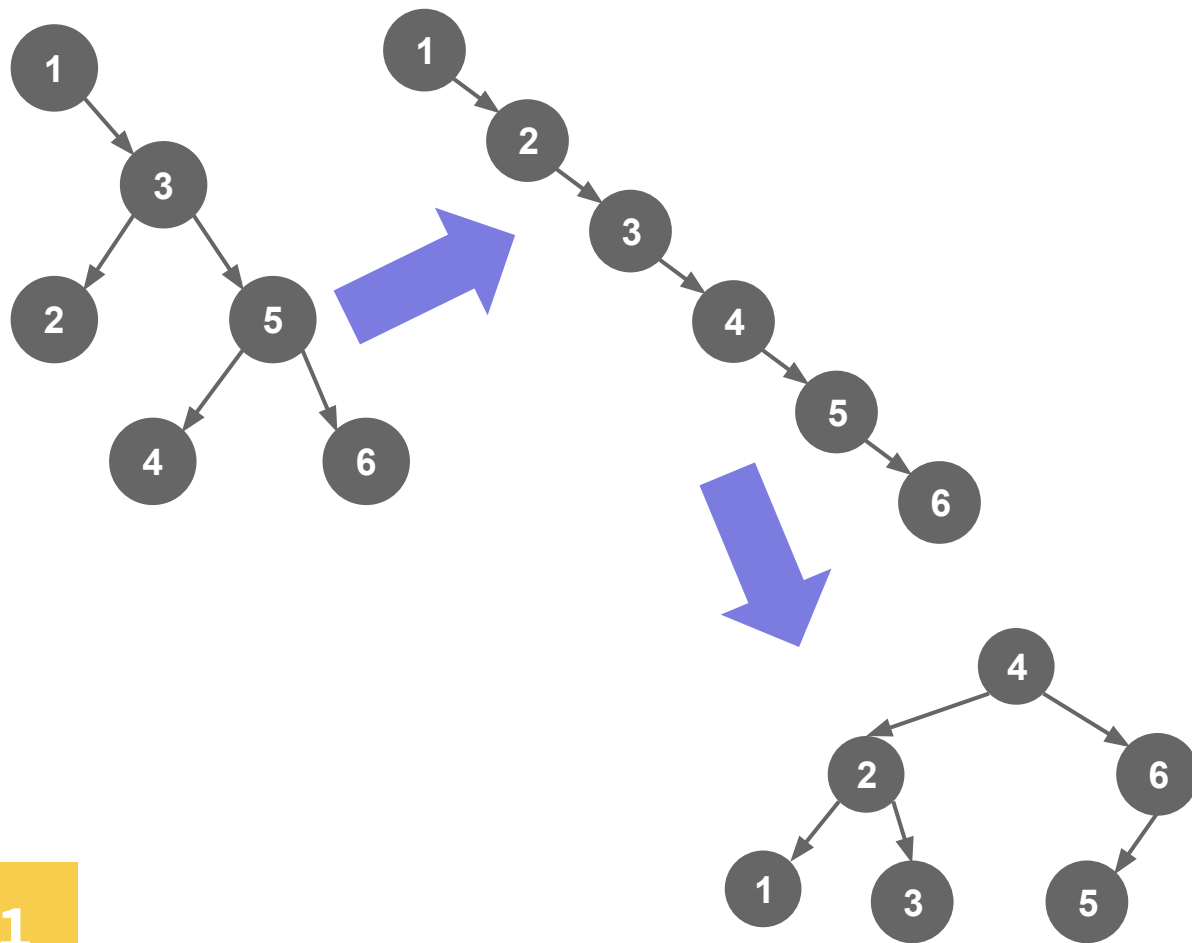


♥ Rotação a direita

♥ Rotação a esquerda

♥ Árvore → backbone

♥ Backbone → árvore



♥ Rotação a direita

♥ Rotação a esquerda

♥ Árvore → backbone

♥ Backbone → árvore

INSERÇÃO

- + Ordenação $O(n)$
- Requer memória(n)
- Destruir e reconstruir

Busca binária

ROTAÇÃO

Rotacionar desbalanceada

Aplica rotações
Transforma em backbone
Aplica rotações
Rotaciona até balancear

Magia?

INSERÇÃO

- + Ordenação $O(n)$
- Requer memória(n)
- Destruir e reconstruir

Busca binária

ROTAÇÃO

- + Economia memória
-

Tecnologia!

ÁRVORES BINÁRIAS



Balanceamento