Análise de Algoritmos

Parte destes slides são adaptações de slides

do Prof. Paulo Feofiloff e do Prof. José Coelho de Pina.

Análise amortizada

Notas de aula de um curso do Robert Tarjan

“*Amortized Analysis Explained*”

por Rebecca Fiebrink, Princeton University

Move to front

Considere uma lista ligada.

Custo do acesso ao i-ésimo elemento: i.

Move to front

Considere uma lista ligada.

Custo do acesso ao i-ésimo elemento: i.

Custo de trocar dois elementos consecutivos de lugar: constante, descontado o tempo de acessar um deles.

Move to front

Considere uma lista ligada.

Custo do acesso ao i-ésimo elemento: i.

Custo de trocar dois elementos consecutivos de lugar:
constante, descontado o tempo de acessar um deles.

Heurística MTF:
ao acessar o elemento x, mova-o para o ínicio da lista.

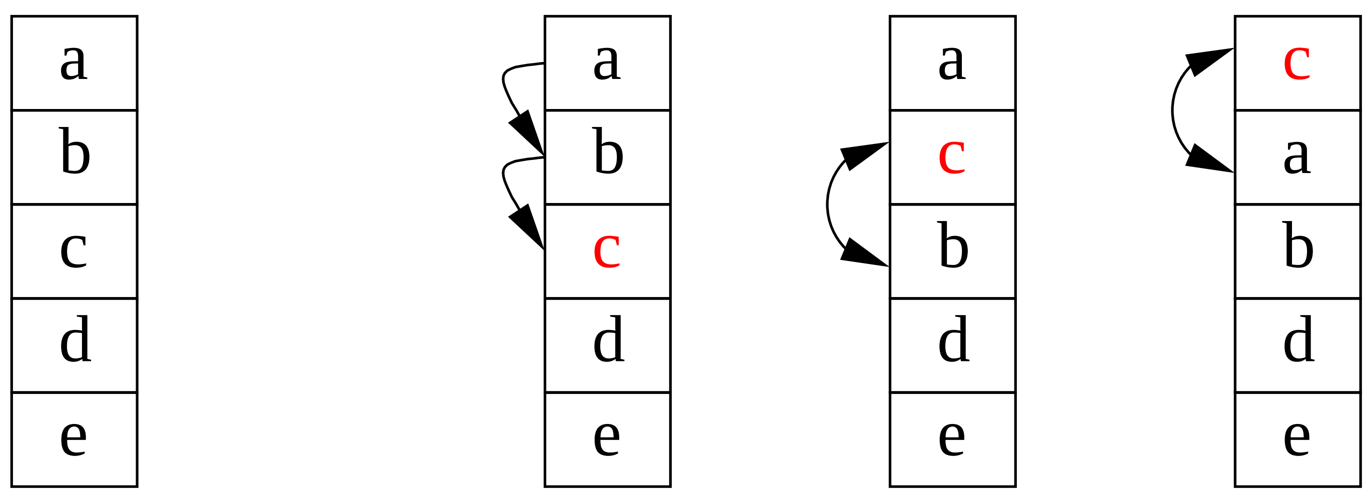
Move to front

Considere uma lista ligada.

Custo do acesso ao i-ésimo elemento: i.

Custo de trocar dois elementos consecutivos de lugar:
constante, descontado o tempo de acessar um deles.

Heurística MTF:
ao acessar o elemento x, mova-o para o ínicio da lista.



Move to front

Considere uma lista ligada.

Custo do acesso ao i-ésimo elemento: i.

Custo de trocar dois elementos consecutivos de lugar:
constante, descontado o tempo de acessar um deles.

Heurística MTF:
ao acessar o elemento x, mova-o para o ínicio da lista.

Com MTF, custo do acesso ao i-ésimo elemento: $2i - 1$.

Move to front

Considere uma lista ligada.

Custo do acesso ao i-ésimo elemento: i.

Custo de trocar dois elementos consecutivos de lugar:
constante, descontado o tempo de acessar um deles.

Heurística MTF:
ao acessar o elemento x, mova-o para o ínicio da lista.

Com MTF, custo do acesso ao i-ésimo elemento: $2i - 1$.

Considere uma sequência de acessos à lista.

Qual é o custo de MTF se comparado
com um algoritmo ótimo de acesso?

Move to front

Lista ligada e sequência de acessos a ela.

Heurística MTF:
ao acessar o elemento x, mova-o para o ínicio da lista.

Acesso ao i-ésimo elemento custa $2i - 1$.

Qual é o custo de MTF se comparado
com um algoritmo ótimo de acesso?

Move to front

Lista ligada e sequência de acessos a ela.

Heurística MTF:
ao acessar o elemento x, mova-o para o ínicio da lista.

Acesso ao i-ésimo elemento custa $2i - 1$.

Qual é o custo de MTF se comparado
com um algoritmo ótimo de acesso?

A: algoritmo arbitrário de acesso à lista.

Análise amortizada:

custo do MTF ≤ 4 vezes o custo de A.

Operações sobre a lista

Acesso e troca com anterior.

Função potencial ϕ: 2 vezes o número de inversões da lista de MTF em relação à lista de A.

Operações sobre a lista

Acesso e troca com anterior.

Função potencial ϕ: 2 vezes o número de inversões da lista de MTF em relação à lista de A.

Seja x o elemento acessado.

Operações sobre a lista

Acesso e troca com anterior.

Função potencial ϕ: 2 vezes o número de inversões da lista de MTF em relação à lista de A.

Seja x o elemento acessado.

Seja k a posição de x na lista de MTF.

Seja i a posição de x na lista de A.

Operações sobre a lista

Acesso e troca com anterior.

Função potencial ϕ: 2 vezes o número de inversões da lista de MTF em relação à lista de A.

Seja x o elemento acessado.

Seja k a posição de x na lista de MTF.

Seja i a posição de x na lista de A.

Suponha que A não troca ninguém de lugar.

Custo pelo acesso a x por MTF: $2k - 1$.

Custo pelo acesso a x por A: i

Custo amortizado do acesso

Função potencial ϕ: 2 vezes o número de inversões da lista de MTF em relação à lista de A.

k: posição de x na lista de MTF.
i: posição de x na lista de A.

Custo amortizado do acesso

Função potencial ϕ: 2 vezes o número de inversões da lista de MTF em relação à lista de A.

k: posição de x na lista de MTF.
i: posição de x na lista de A.

Em MTF, depois do acesso,
pares com x e um dos k elementos trocaram de posição.

Custo amortizado do acesso

Função potencial ϕ: 2 vezes o número de inversões da lista de MTF em relação à lista de A.

k: posição de x na lista de MTF.
i: posição de x na lista de A.

Em MTF, depois do acesso, pares com x e um dos k elementos trocaram de posição.

Existem $i - 1$ elementos na lista de A na frente de x.

Custo amortizado do acesso

Função potencial ϕ: 2 vezes o número de inversões da lista de MTF em relação à lista de A.

k: posição de x na lista de MTF.
i: posição de x na lista de A.

Em MTF, depois do acesso,
pares com x e um dos k elementos trocaram de posição.

Existem $i - 1$ elementos na lista de A na frente de x.

$\leq \min\{k-1, i-1\}$ inversões a mais depois do acesso

Custo amortizado do acesso

Função potencial ϕ: 2 vezes o número de inversões da lista de MTF em relação à lista de A.

k: posição de x na lista de MTF.
i: posição de x na lista de A.

Em MTF, depois do acesso,
pares com x e um dos k elementos trocaram de posição.

Existem $i - 1$ elementos na lista de A na frente de x.

$\leq \min\{k-1, i-1\}$ inversões a mais depois do acesso

$\geq (k-1) - \min\{k-1, i-1\}$ inversões a menos

Custo amortizado do acesso

Função potencial ϕ: 2 vezes o número de inversões da lista de MTF em relação à lista de A.

k: posição de x na lista de MTF.
i: posição de x na lista de A.

Em MTF, depois do acesso, pares com x e um dos k elementos trocaram de posição.

Existem $i-1$ elementos na lista de A na frente de x.

$\leq \min\{k-1, i-1\}$ inversões a mais depois do acesso

$\geq (k-1) - \min\{k-1, i-1\}$ inversões a menos

Então $\Delta\Phi \leq 2(2\min\{k-1, i-1\} - (k-1))$.

Custo amortizado do acesso

Função potencial ϕ: 2 vezes o número de inversões da lista de MTF em relação à lista de A.

$\Delta\Phi \leq 4\min\{k-1, i-1\} - 2(k-1)$.

Custo amortizado do acesso

Função potencial ϕ: 2 vezes o número de inversões da lista de MTF em relação à lista de A.

$\Delta\Phi \leq 4\min\{k-1, i-1\} - 2(k-1)$.

Custo amortizado:

$$
\begin{aligned}
\hat{c} &= c + \Delta\Phi \\
&\leq 2k - 1 + 4\min\{k-1, i-1\} - 2(k-1) \\
&\leq 1 + 4\min\{k-1, i-1\} \\
&\leq 4i.
\end{aligned}
$$

Custo amortizado do acesso

Função potencial ϕ: 2 vezes o número de inversões da lista de MTF em relação à lista de A.

$\Delta\Phi \leq 4\min\{k-1, i-1\} - 2(k-1)$.

Custo amortizado:

$$\begin{aligned}
\hat{c} &= c + \Delta\Phi \\
&\leq 2k - 1 + 4\min\{k-1, i-1\} - 2(k-1) \\
&\leq 1 + 4\min\{k-1, i-1\} \\
&\leq 4i.
\end{aligned}$$

Logo o custo amortizado por acesso de A é no máximo 4.

Custo amortizado

Função potencial ϕ: 2 vezes o número de inversões da lista de MTF em relação à lista de A.

O custo amortizado por acesso é menor que 4 em relação ao custo de A... se A não fizer trocas...

Custo amortizado

Função potencial ϕ: 2 vezes o número de inversões da lista de MTF em relação à lista de A.

O custo amortizado por acesso é menor que 4 em relação ao custo de A... se A não fizer trocas...

E se A fizer trocas também?

Custo amortizado

Função potencial ϕ: 2 vezes o número de inversões da lista de MTF em relação à lista de A.

O custo amortizado por acesso é menor que 4 em relação ao custo de A... se A não fizer trocas...

E se A fizer trocas também?

Com uma troca, o custo de A sobe de 1.

Custo amortizado

Função potencial ϕ: 2 vezes o número de inversões da lista de MTF em relação à lista de A.

O custo amortizado por acesso é menor que 4 em relação ao custo de A... se A não fizer trocas...

E se A fizer trocas também?

Com uma troca, o custo de A sobe de 1.

E o potencial, como se altera? Sobe ou desce de 2.

Custo amortizado

Função potencial ϕ: 2 vezes o número de inversões da lista de MTF em relação à lista de A.

O custo amortizado por acesso é menor que 4 em relação ao custo de A... se A não fizer trocas...

E se A fizer trocas também?

Com uma troca, o custo de A sobe de 1.

E o potencial, como se altera? Sobe ou desce de 2.

Ou seja, o custo de A é $i + t$,
onde t é o número de trocas de A após o acesso de x,
e $\Delta\Phi \leq 4 \min\{k-1, i-1\} - 2(k-1) + 2t$.

Custo amortizado

Função potencial ϕ: 2 vezes o número de inversões da lista de MTF em relação à lista de A.

O custo amortizado por acesso é menor que 4 em relação ao custo de A... se A não fizer trocas...

E se A fizer trocas também?

Com uma troca, o custo de A sobe de 1.

E o potencial, como se altera? Sobe ou desce de 2.

Ou seja, o custo de A é $i + t$,
onde t é o número de trocas de A após o acesso de x,
e $\Delta\Phi \leq 4\min\{k-1, i-1\} - 2(k-1) + 2t$.

Custo amortizado: $\hat{c} \leq 4i + 2t \leq 4(i+t)$.

Custo amortizado

Função potencial ϕ: 2 vezes o número de inversões da lista de MTF em relação à lista de A.

O custo amortizado por acesso é menor que 4 em relação ao custo de A... se A não fizer trocas...

E se A fizer trocas também?

Com uma troca, o custo de A sobe de 1.

E o potencial, como se altera? Sobe ou desce de 2.

Ou seja, o custo de A é $i + t$,
onde t é o número de trocas de A após o acesso de x,
e $\Delta\Phi \leq 4\min\{k-1, i-1\} - 2(k-1) + 2t$.

Custo amortizado: $\hat{c} \leq 4i + 2t \leq 4(i+t)$.

Custo amortizado por operação de A é ≤ 4.

Splay trees

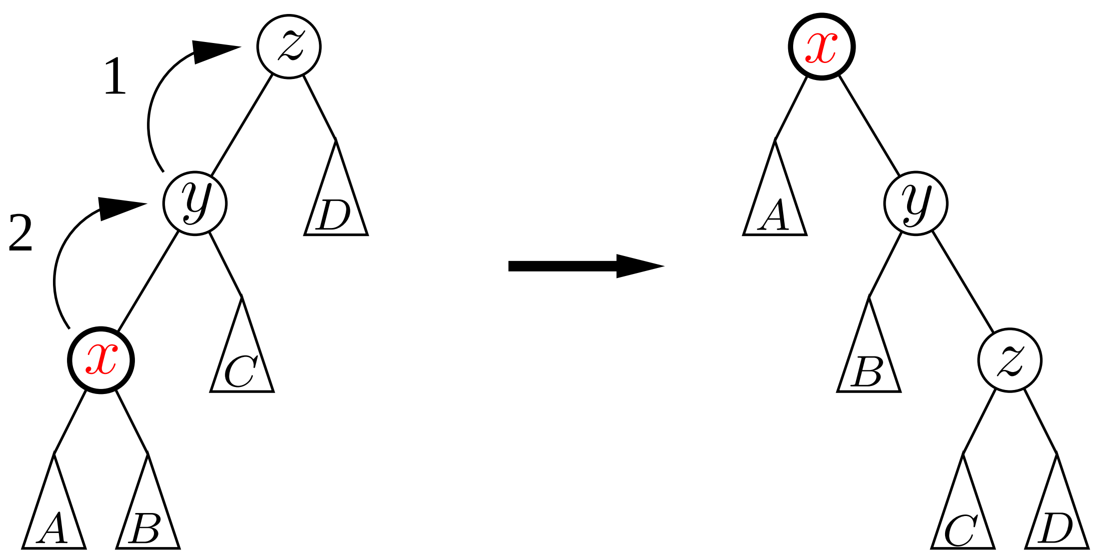
ABB: árvore de busca binária

Operação SPLAY(x, S), onde S é uma splay tree:
quando x é acessado, move-se x para a raiz por rotações.
Para a análise, olhamos duas rotações de cada vez.

Splay trees

ABB: árvore de busca binária

Operação SPLAY(x, S), onde S é uma splay tree:
quando x é acessado, move-se x para a raiz por rotações.
Para a análise, olhamos duas rotações de cada vez.

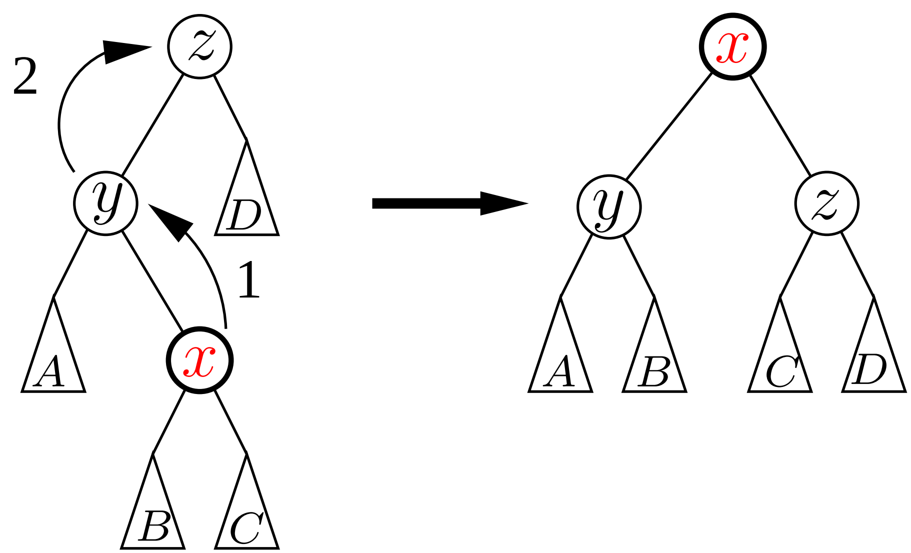


Acima, o rr splay step.

Splay trees

ABB: árvore de busca binária

Operação SPLAY(x, S), onde S é uma splay tree:
quando x é acessado, move-se x para a raiz por rotações.
Para a análise, olhamos duas rotações de cada vez.

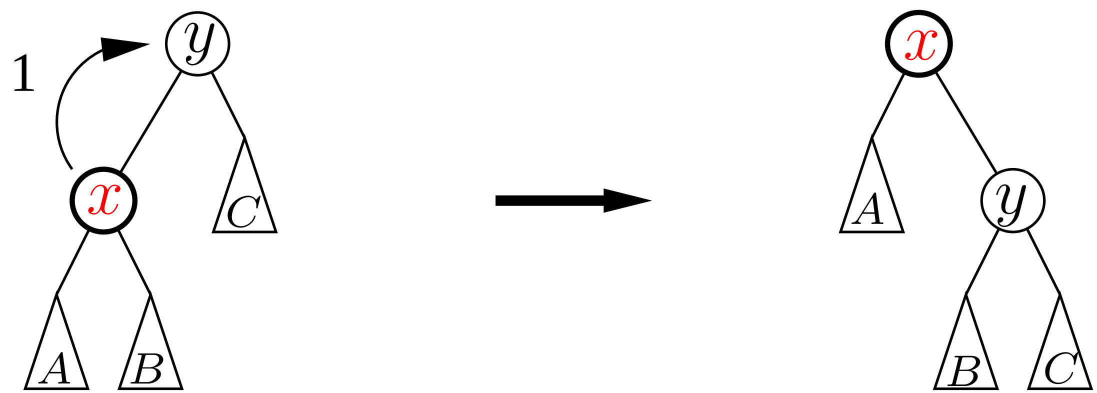


Acima, o lr splay step.

Splay trees

ABB: árvore de busca binária

Operação SPLAY(x, S), onde S é uma splay tree:
quando x é acessado, move-se x para a raiz por rotações.
Para a análise, olhamos duas rotações de cada vez.
Eventualmente no final uma rotação simples é necessária.

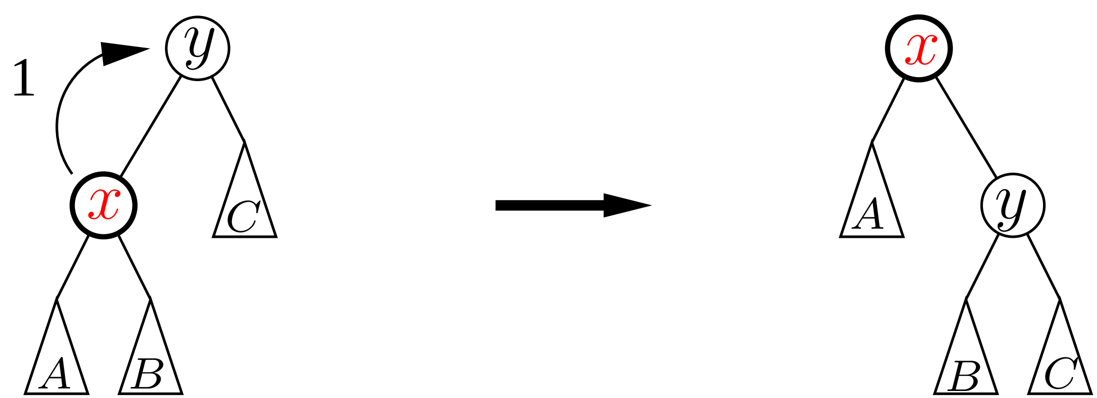


Acima, o r splay step.

Splay trees

ABB: árvore de busca binária

Operação SPLAY(x, S), onde S é uma splay tree:
quando x é acessado, move-se x para a raiz por rotações.
Para a análise, olhamos duas rotações de cada vez.
Eventualmente no final uma rotação simples é necessária.



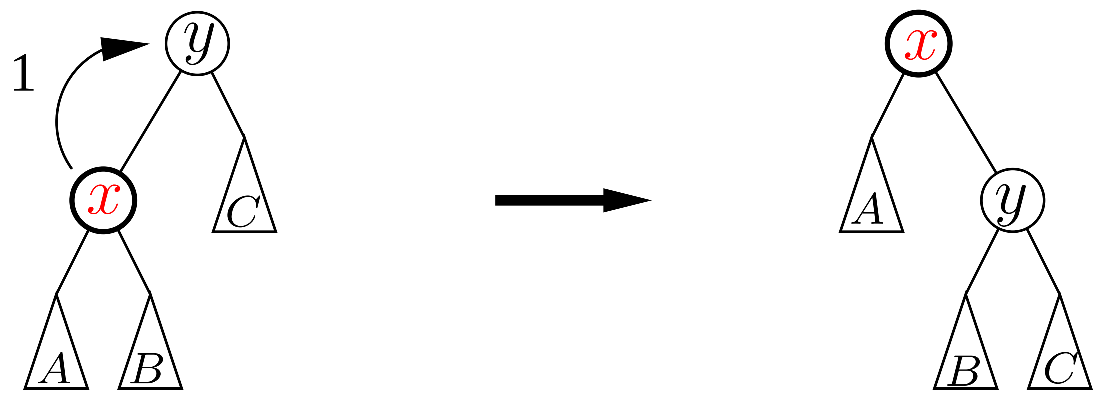
Acima, o r splay step.

Além destes, o l splay, o rl splay e o ll splay step.

Splay trees

ABB: árvore de busca binária

Operação SPLAY(x, S), onde S é uma splay tree:
quando x é acessado, move-se x para a raiz por rotações.
Para a análise, olhamos duas rotações de cada vez.
Eventualmente no final uma rotação simples é necessária.



Acima, o r splay step.

Além destes, o l splay, o rl splay e o ll splay step.

Splay steps são realizados até que x seja raiz.

Splay trees

ABB: árvore de busca binária

Operação SPLAY(x, S), onde S é uma splay tree:
quando x é acessado, move-se x para a raiz por rotações.

Splay trees

ABB: árvore de busca binária

Operação SPLAY(x, S), onde S é uma splay tree:
quando x é acessado, move-se x para a raiz por rotações.

Custo de pior caso do SPLAY: $\Theta(n)$,
onde n é o número de elementos na splay tree.

Splay trees

ABB: árvore de busca binária

Operação SPLAY(x, S), onde S é uma splay tree:
quando x é acessado, move-se x para a raiz por rotações.

Custo de pior caso do SPLAY: $\Theta(n)$,
onde n é o número de elementos na splay tree.

Queremos mostrar que uma splay tree tem um comportamento mais parecido com o de uma ABBB.

Splay trees

ABB: árvore de busca binária

Operação SPLAY(x, S), onde S é uma splay tree:
quando x é acessado, move-se x para a raiz por rotações.

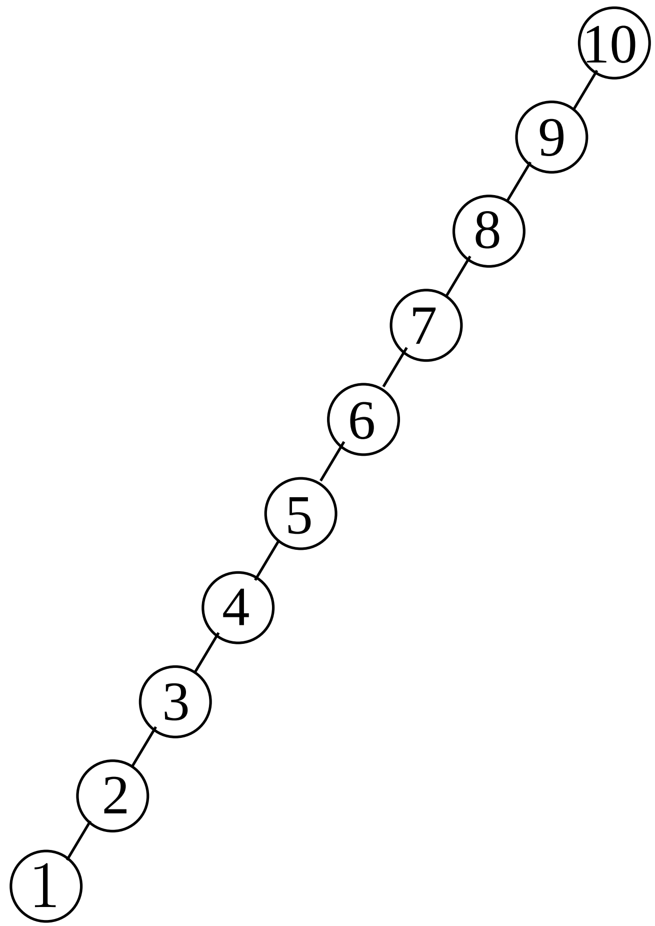
Custo de pior caso do SPLAY: $\Theta(n)$,
onde n é o número de elementos na splay tree.

Queremos mostrar que uma splay tree tem um comportamento mais parecido com o de uma ABBB.

(ABBB: ABB balanceada)

Splay trees: pior caso

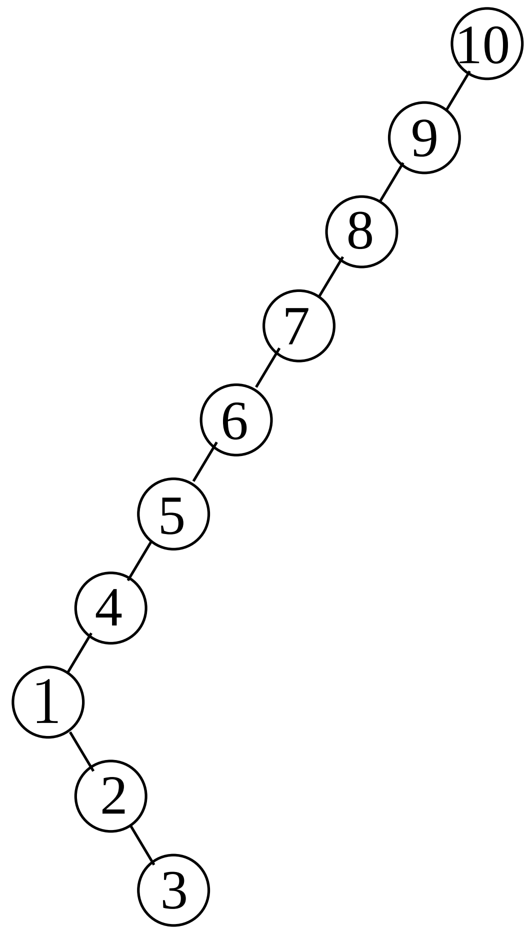
SPLAY$(1, S)$



Splay trees: pior caso

SPLAY$(1, S)$

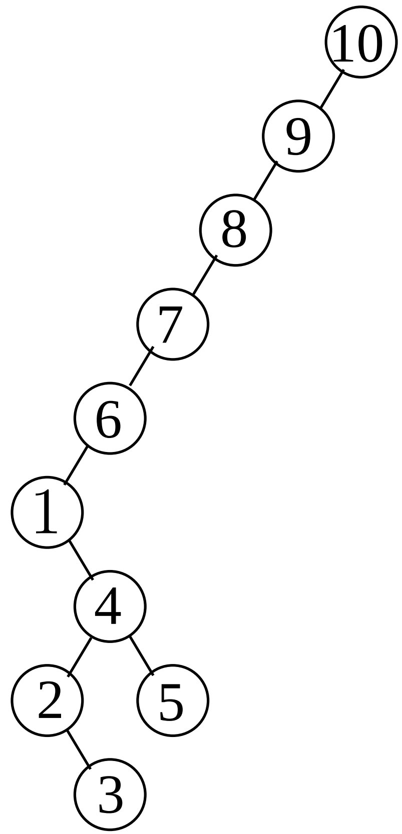
primeiro splay step



Splay trees: pior caso

SPLAY$(1, S)$

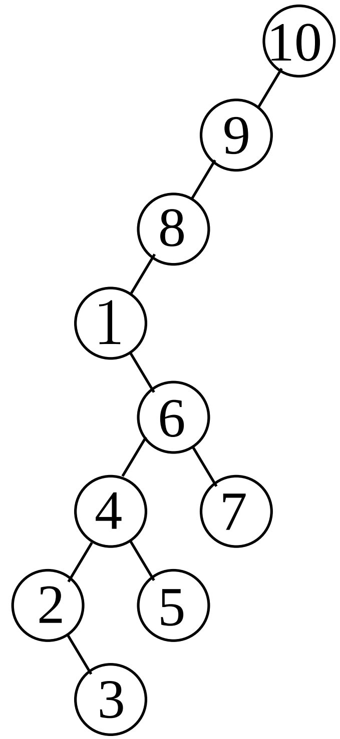
segundo splay step



Splay trees: pior caso

SPLAY$(1, S)$

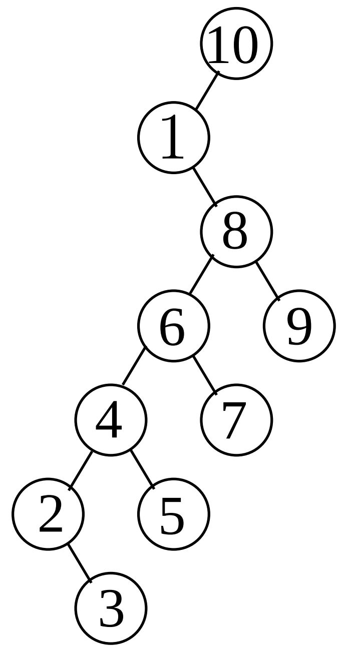
terceiro splay step



Splay trees: pior caso

SPLAY$(1, S)$

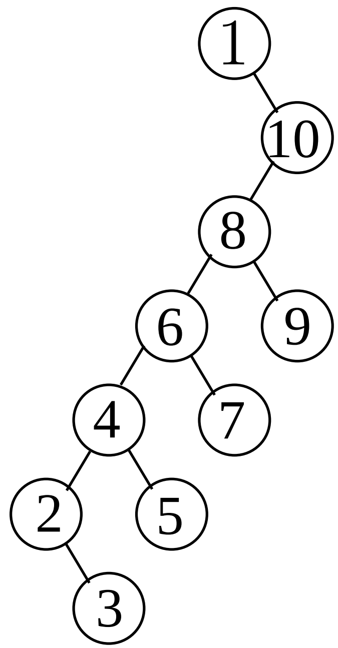
quarto splay step



Splay trees: pior caso

SPLAY$(1, S)$

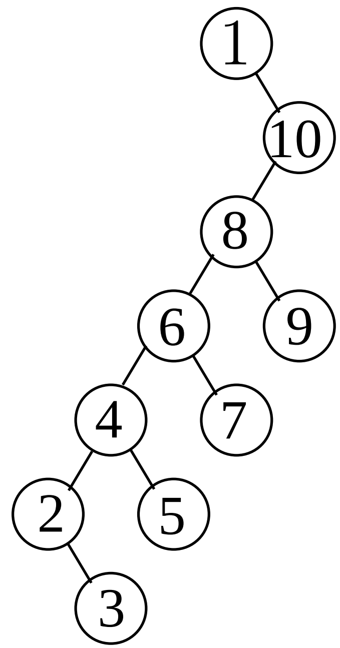
quinto splay step



Splay trees: pior caso

SPLAY$(1, S)$

quinto splay step



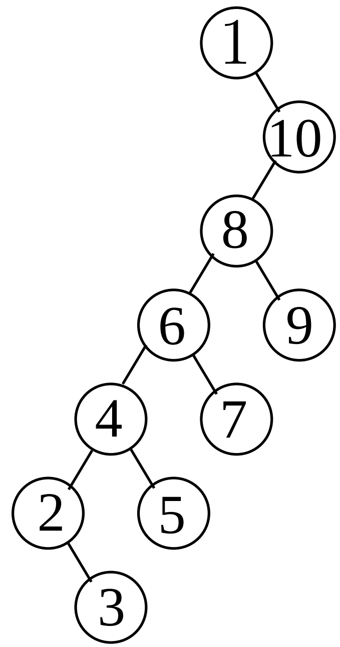
Total de rotações: 9

Em geral, é $\Theta(n)$, onde n é o número de nós.

Agora, o maior custo de um SPLAY nesta árvore é 6.

Splay trees: mais um exemplo

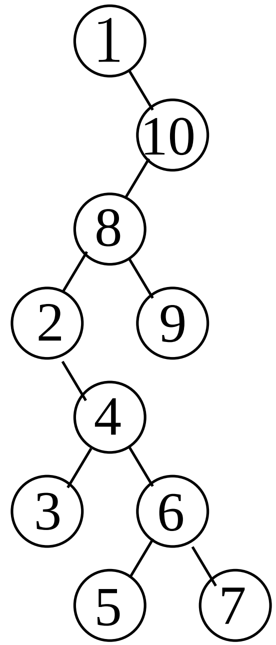
SPLAY$(2, S)$



Splay trees: mais um exemplo

SPLAY$(2, S)$

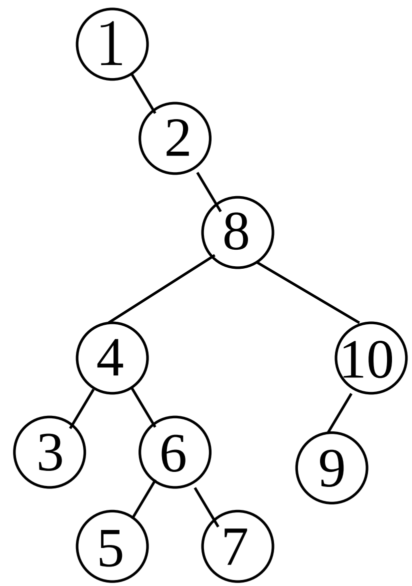
primeiro splay step



Splay trees: mais um exemplo

SPLAY$(2, S)$

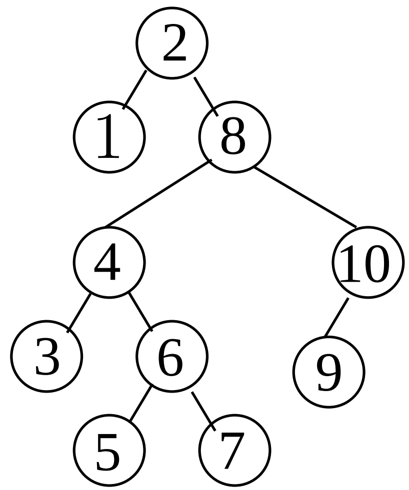
segundo splay step



Splay trees: mais um exemplo

SPLAY$(2, S)$

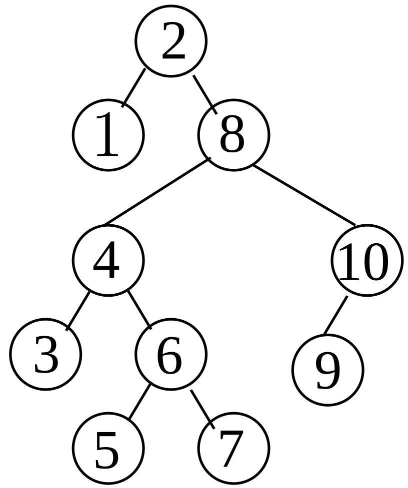
terceiro splay step



Splay trees: mais um exemplo

SPLAY$(2, S)$

terceiro splay step



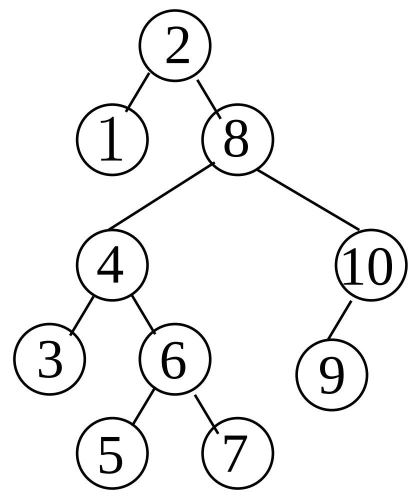
Total de rotações: 5

Árvore bem mais balanceada após estes dois SPLAYs.

Splay trees: mais um exemplo

SPLAY$(2, S)$

terceiro splay step



Total de rotações: 5

Árvore bem mais balanceada após estes dois SPLAYs.

Custo: número de rotações.

Splay trees

S: splay tree

Splay trees

S: splay tree

$S_i(x)$: subárvore de S enraizada em x no instante i
$s_i(x) = |S_i(x)|$

Splay trees

S: splay tree

$S_i(x)$: subárvore de S enraizada em x no instante i
$s_i(x) = |S_i(x)|$

Tome $r_i(x) = \lg s_i(x)$.

$r_i(x)$ indica o potencial local no nó x.

Splay trees

S: splay tree

$S_i(x)$: subárvore de S enraizada em x no instante i
$s_i(x) = |S_i(x)|$

Tome $r_i(x) = \lg s_i(x)$.

$r_i(x)$ indica o potencial local no nó x.

Seja $\Phi_i = \sum_x r_i(x)$.

Splay trees

S: splay tree

$S_i(x)$: subárvore de S enraizada em x no instante i
$s_i(x) = |S_i(x)|$

Tome $r_i(x) = \lg s_i(x)$.

$r_i(x)$ indica o potencial local no nó x.

Seja $\Phi_i = \sum_x r_i(x)$.

Mostraremos que o custo amortizado por SPLAY é $\mathrm{O}(\lg n)$.

Splay trees

S: splay tree

$S_i(x)$: subárvore de S enraizada em x no instante i
$s_i(x) = |S_i(x)|$

Tome $r_i(x) = \lg s_i(x)$.

$r_i(x)$ indica o potencial local no nó x.

Seja $\Phi_i = \sum_x r_i(x)$.

Mostraremos que o custo amortizado por SPLAY é $\mathrm{O}(\lg n)$.

Análise amortizada dos splay steps.

Splay trees

S: splay tree

$S_i(x)$: subárvore de S enraizada em x no instante i
$s_i(x) = |S_i(x)|$

Tome $r_i(x) = \lg s_i(x)$.

$r_i(x)$ indica o potencial local no nó x.

Seja $\Phi_i = \sum_x r_i(x)$.

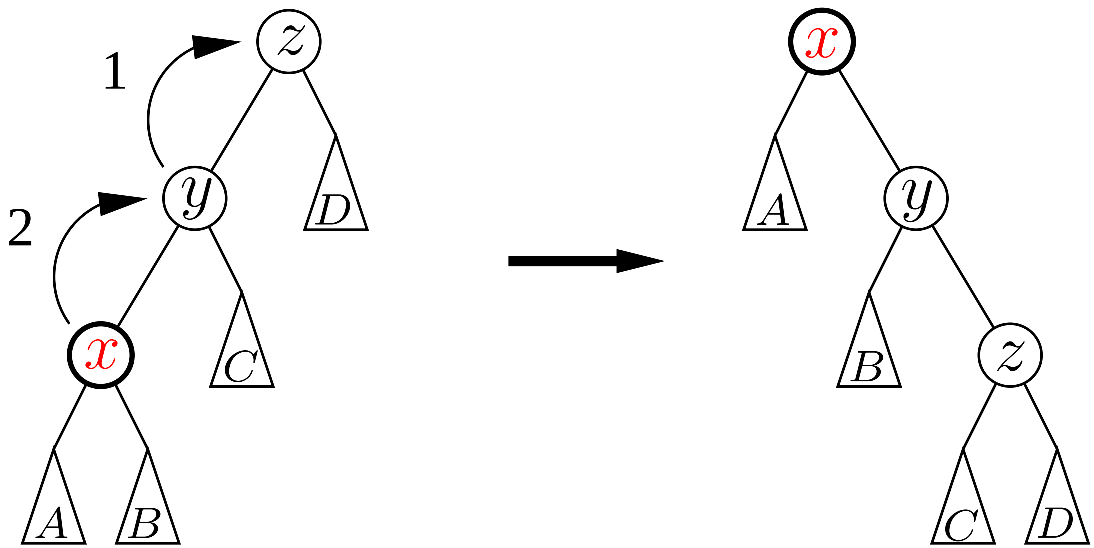
Mostraremos que o custo amortizado por SPLAY é $\mathrm{O}(\lg n)$.

Análise amortizada dos splay steps.

x participa de todos os splay steps de SPLAY(x, S).

Análise amortizada dos splay steps

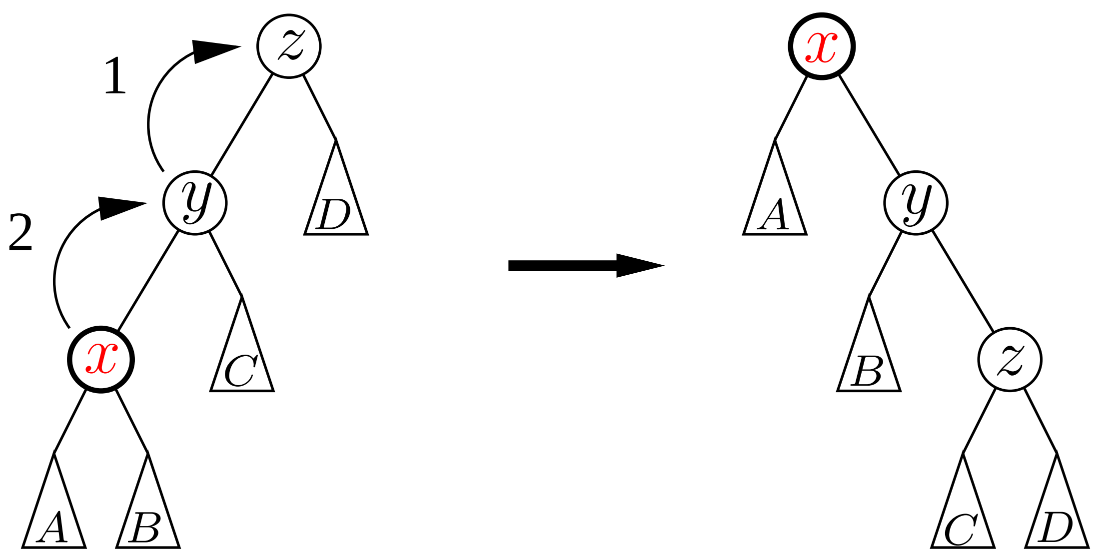
Caso do rr splay step.



Custo real: 2

Análise amortizada dos splay steps

Caso do rr splay step.



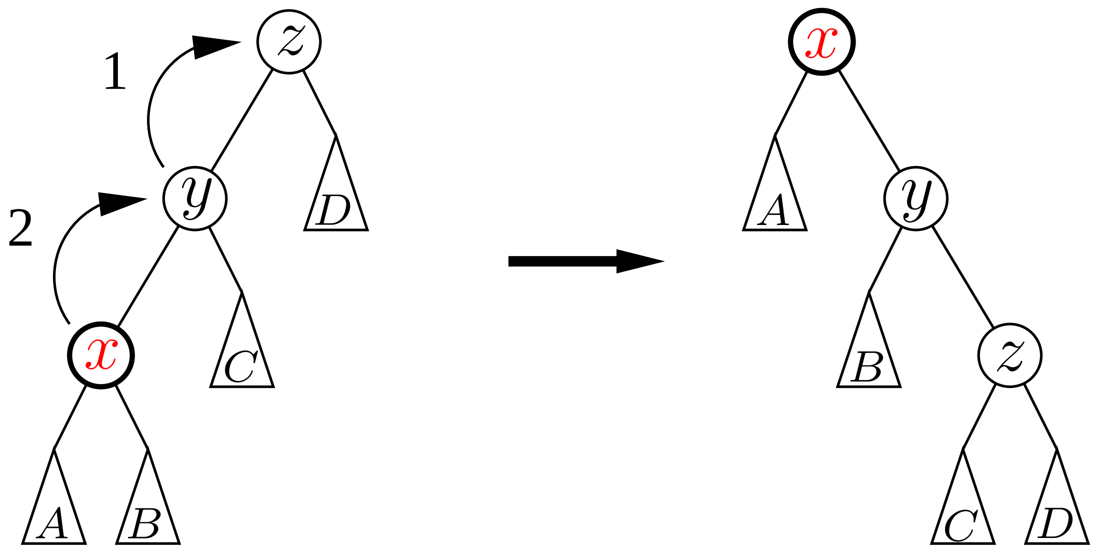
Custo real: 2

Alteração no potencial:

$$\Phi_i - \Phi_{i-1} = \sum_w r_i(w) - \sum_w r_{i-1}(w) = \sum_w (r_i(w) - r_{i-1}(w))$$

Análise amortizada dos splay steps

Caso do rr splay step.



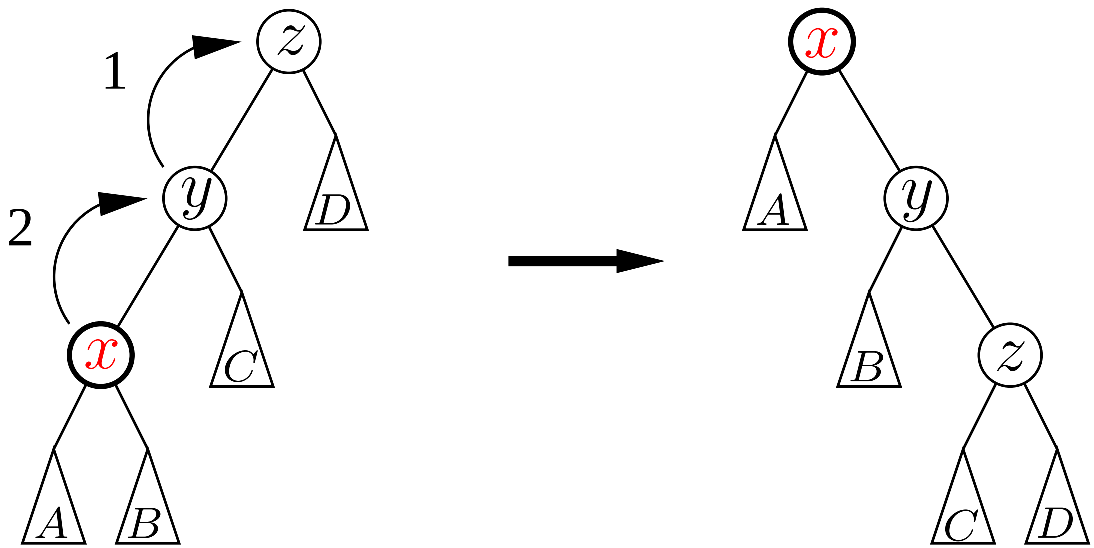
Custo real: 2

Alteração no potencial:

$$\begin{aligned}\Phi_i - \Phi_{i-1} &= \sum_w r_i(w) - \sum_w r_{i-1}(w) = \sum_w (r_i(w) - r_{i-1}(w)) \\ &= (r_i(x) - r_{i-1}(x)) + (r_i(y) - r_{i-1}(y)) + (r_i(z) - r_{i-1}(z))\end{aligned}$$

Análise amortizada dos splay steps

Caso do rr splay step.



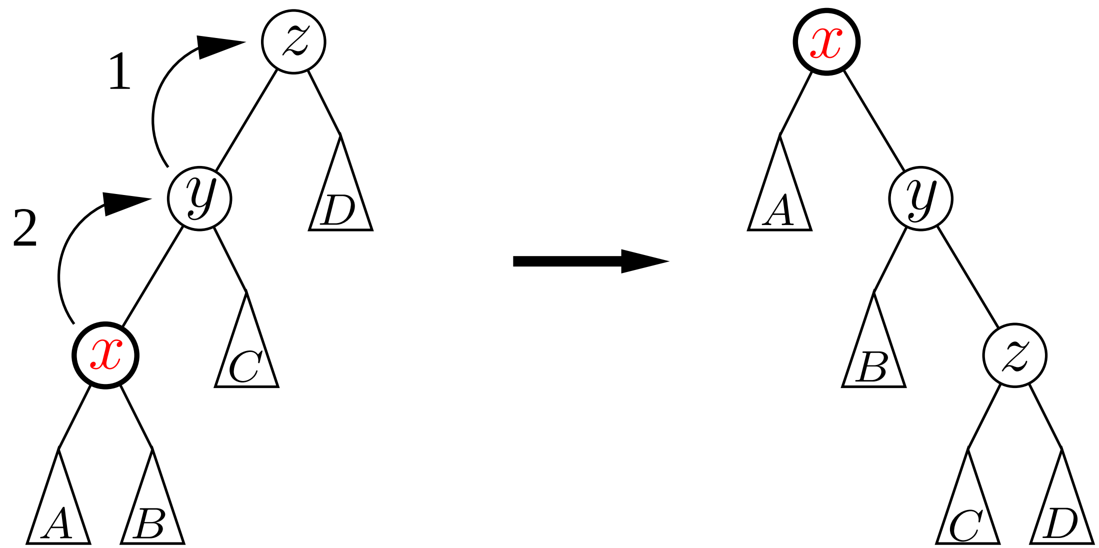
Custo real: 2

Alteração no potencial:

$$
\begin{aligned}
\Phi_i - \Phi_{i-1} &= \textstyle\sum_w r_i(w) - \sum_w r_{i-1}(w) = \sum_w (r_i(w) - r_{i-1}(w)) \\
&= (r_i(x) - r_{i-1}(x)) + (r_i(y) - r_{i-1}(y)) + (r_i(z) - r_{i-1}(z)) \\
&= (r_i(y) + r_i(z)) - (r_{i-1}(x) + r_{i-1}(y))
\end{aligned}
$$

Análise amortizada dos splay steps

Caso do rr splay step.



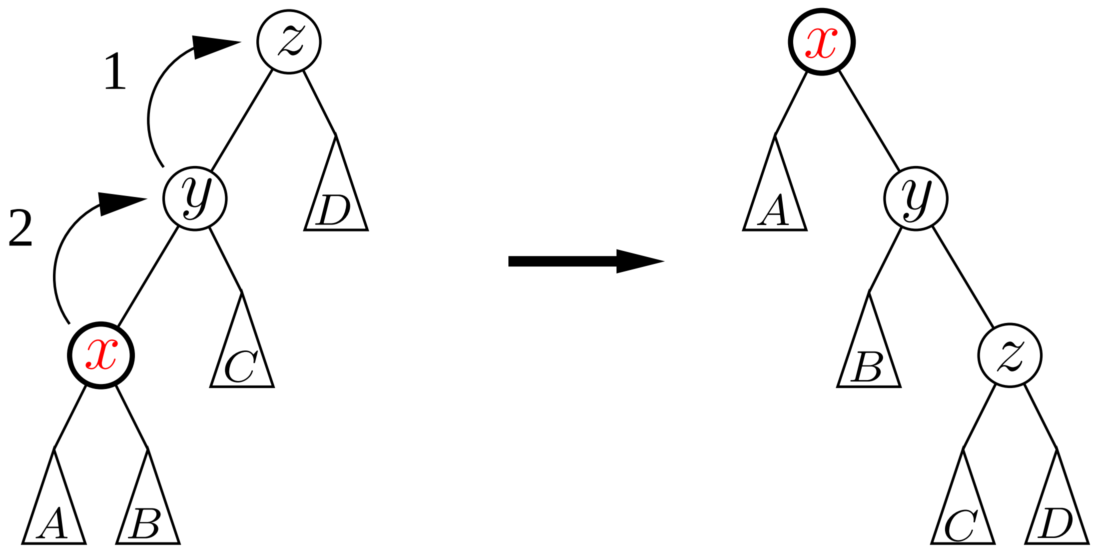
Custo real: 2

Alteração no potencial:

$$
\begin{aligned}
\Phi_i - \Phi_{i-1} &= \sum_w r_i(w) - \sum_w r_{i-1}(w) = \sum_w (r_i(w) - r_{i-1}(w)) \\
&= (r_i(x) - r_{i-1}(x)) + (r_i(y) - r_{i-1}(y)) + (r_i(z) - r_{i-1}(z)) \\
&= (r_i(y) + r_i(z)) - (r_{i-1}(x) + r_{i-1}(y)) \\
&\leq r_i(y) + r_i(z) - 2\, r_{i-1}(x)
\end{aligned}
$$

Análise amortizada dos splay steps

Caso do rr splay step.

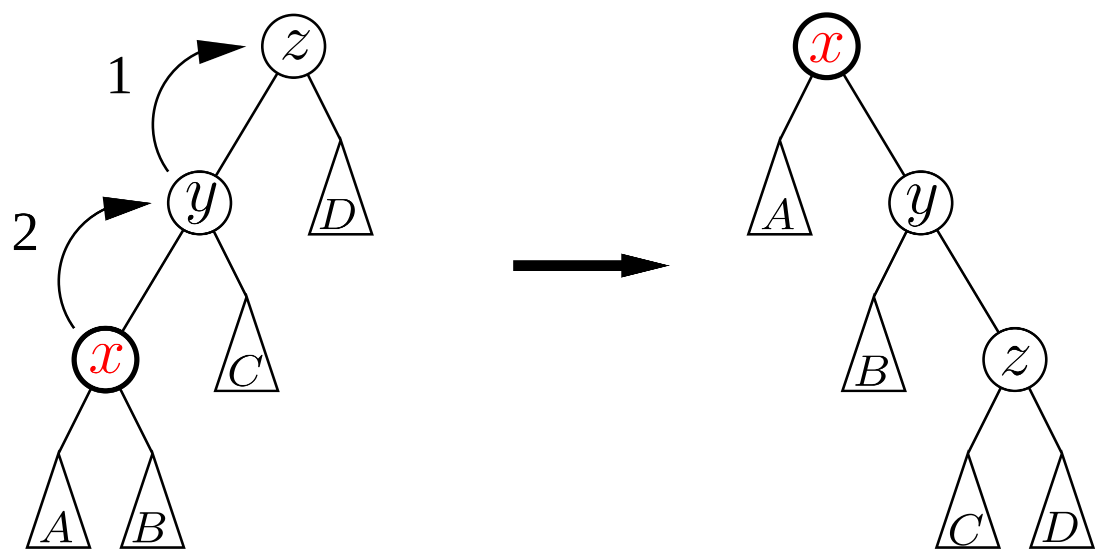


Custo real: 2

Alteração no potencial: $\Phi_i - \Phi_{i-1} \leq r_i(y) + r_i(z) - 2r_{i-1}(x)$

Análise amortizada dos splay steps

Caso do rr splay step.



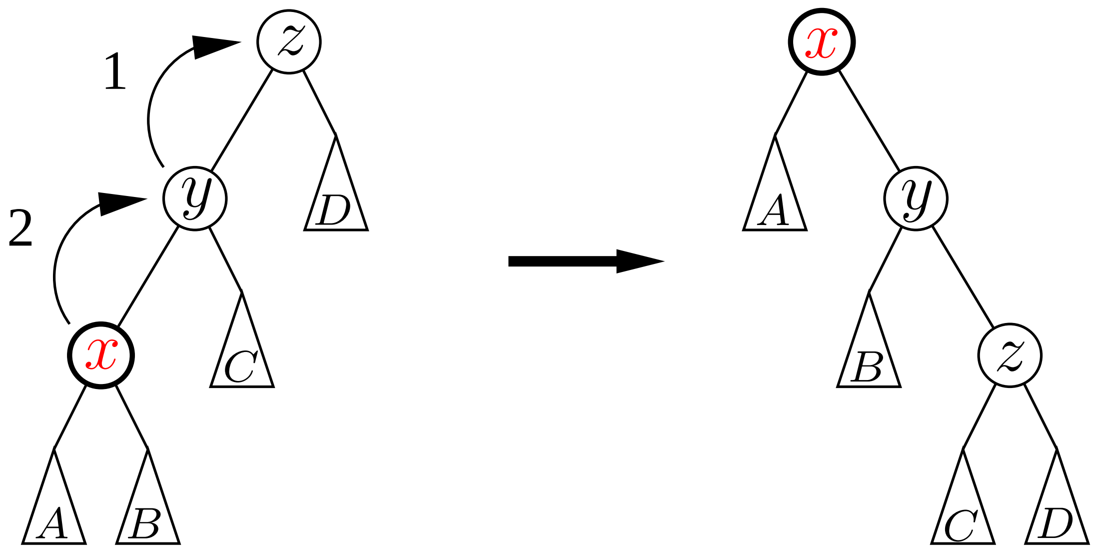
Custo real: 2

Alteração no potencial: $\Phi_i - \Phi_{i-1} \leq r_i(y) + r_i(z) - 2r_{i-1}(x)$

Custo amortizado: $\hat{c}_i \leq 2 + r_i(y) + r_i(z) - 2r_{i-1}(x)$

Análise amortizada dos splay steps

Caso do rr splay step.



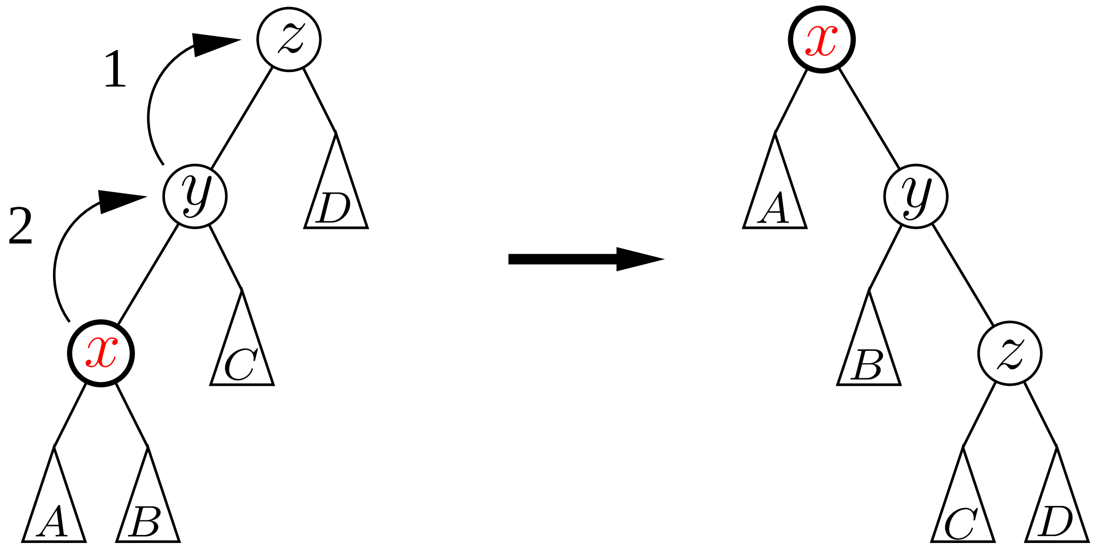
Custo real: 2

Alteração no potencial: $\Phi_i - \Phi_{i-1} \leq r_i(y) + r_i(z) - 2r_{i-1}(x)$

Custo amortizado: $\hat{c}_i \leq 2 + r_i(y) + r_i(z) - 2r_{i-1}(x)$

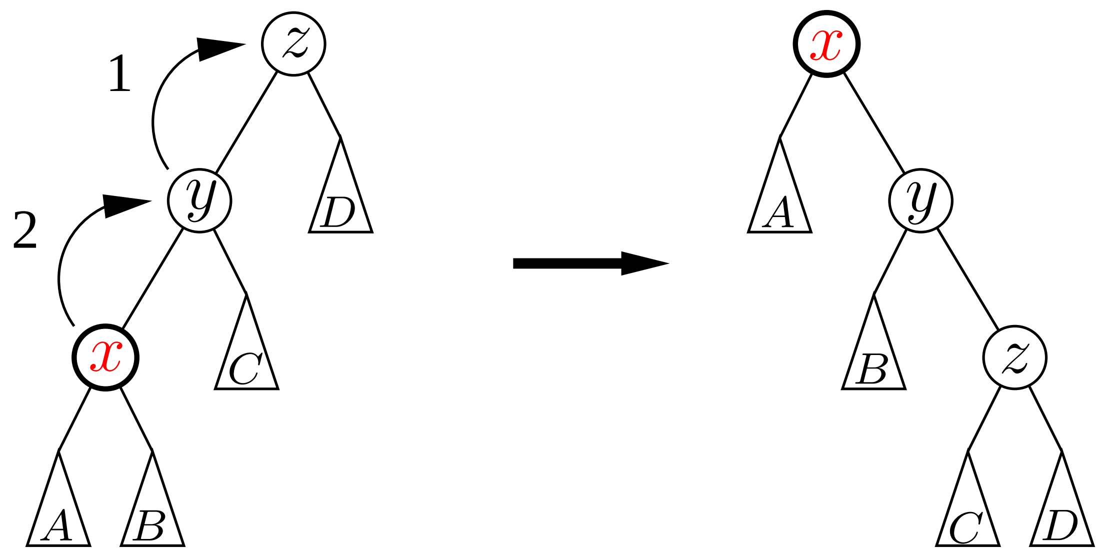
Queremos uma delimitação que dependa apenas de x.

Caso do rr splay step



Temos que $\frac{\log a + \log b}{2} \leq \log(\frac{a+b}{2})$. Logo

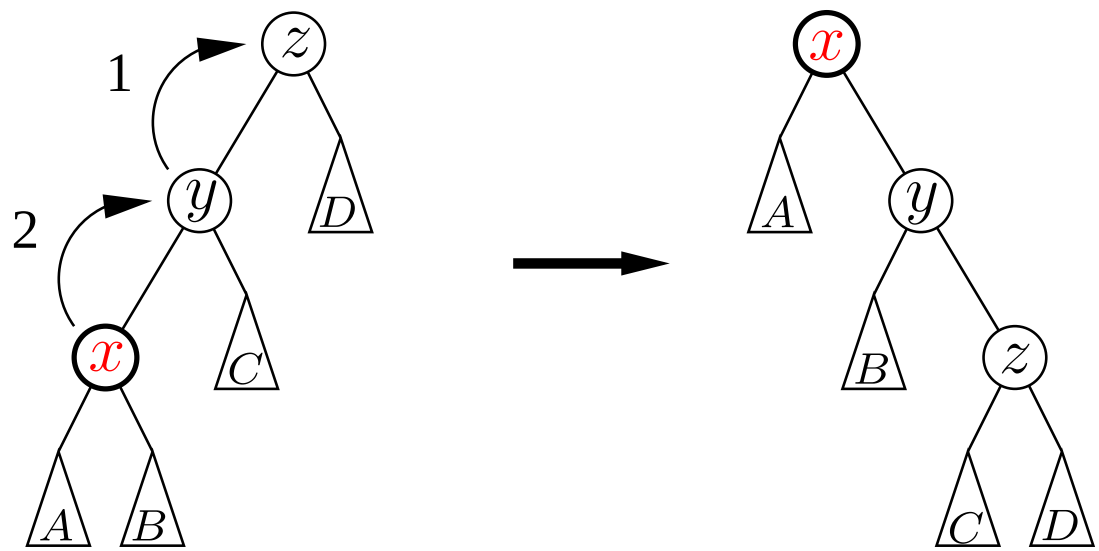
Caso do rr splay step



Temos que $\frac{\log a + \log b}{2} \leq \log(\frac{a+b}{2})$. Logo

$$
\begin{aligned}
r_{i-1}(x) + r_i(z) &= \lg s_{i-1}(x) + \log s_i(z) \\
&\leq 2 \lg \left(\frac{s_{i-1}(x) + s_i(z)}{2}\right) \\
&< 2 \lg \left(\frac{s_i(x)}{2}\right).
\end{aligned}
$$

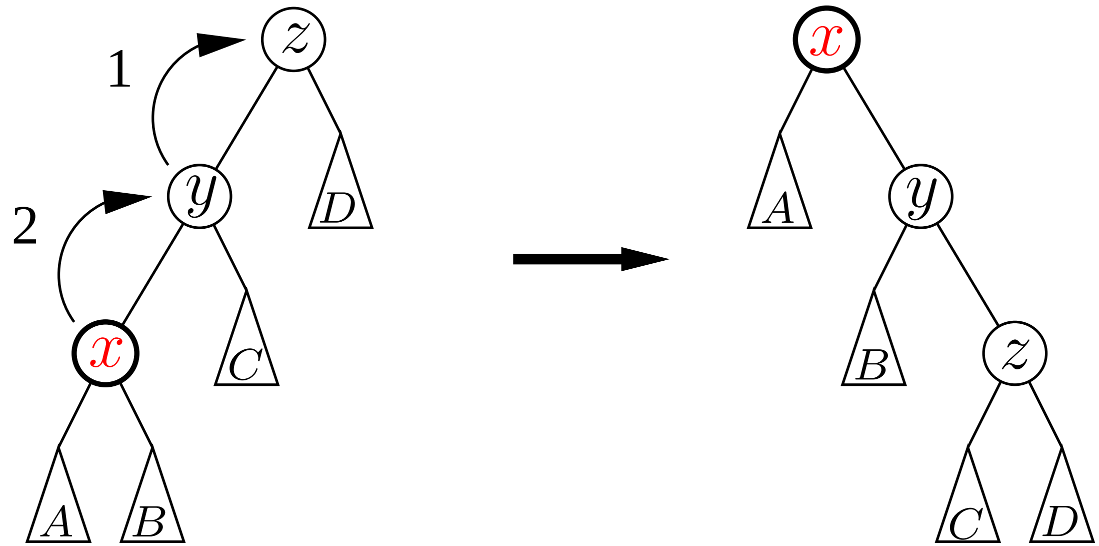
Caso do rr splay step



Temos que $\frac{\log a + \log b}{2} \leq \log(\frac{a+b}{2})$. Logo

$$r_{i-1}(x) + r_i(z) \; < \; 2 \lg s_i(x) - 2 \; = \; 2\, r_i(x) - 2.$$

Caso do rr splay step



Temos que $\frac{\log a + \log b}{2} \leq \log(\frac{a+b}{2})$. Logo

$$r_{i-1}(x) + r_i(z) \; < \; 2 \lg s_i(x) - 2 \; = \; 2\, r_i(x) - 2.$$

Então

$$\begin{aligned}
\hat{c}_i \quad & \leq \quad 2 + r_i(y) + r_i(z) - 2r_{i-1}(x) \\
& < \quad 2 + r_i(x) + (2\, r_i(x) - r_{i-1}(x) - 2) - 2r_{i-1}(x) \\
& < \quad 3(r_i(x) - r_{i-1}(x)).
\end{aligned}$$

Análise amortizada dos splay steps

Analogamente podemos mostrar que

$$\hat{c}_i < 3(r_i(x) - r_{i-1}(x))$$

para rl splay steps, lr splay steps, e ll splay steps.

Análise amortizada dos splay steps

Analogamente podemos mostrar que

$$\hat{c}_i < 3(r_i(x) - r_{i-1}(x))$$

para rl splay steps, lr splay steps, e ll splay steps.

Para l splay steps e r splay steps, vale que

$$\hat{c}_i \leq 3(r_i(x) - r_{i-1}(x)) + 1.$$

Análise amortizada dos splay steps

Analogamente podemos mostrar que

$$\hat{c}_i < 3(r_i(x) - r_{i-1}(x))$$

para rl splay steps, lr splay steps, e ll splay steps.

Para l splay steps e r splay steps, vale que

$$\hat{c}_i \leq 3(r_i(x) - r_{i-1}(x)) + 1.$$

Se m é o número de splay steps e n o número de nós,

$$\begin{aligned} \sum_{i=1}^{m} \hat{c}_i &\leq 3 \sum_{i=1}^{m} (r_i(x) - r_{i-1}(x)) + 1 \\ &= 3(r_m(x) - r_0(x)) + 1 \\ &= 3 \lg n + 1. \end{aligned}$$

Análise amortizada dos splay steps

Lembre-se que $\Phi_i = \sum_x r_i(x)$.

Ou seja, $\Phi_i \geq 0$ para todo i.

Análise amortizada dos splay steps

Lembre-se que $\Phi_i = \sum_x r_i(x)$.

Ou seja, $\Phi_i \geq 0$ para todo i.

Então vale que o custo do SPLAY(x, S) é

$$\sum_{i=1}^{m} c_i \;=\; \sum_{i=1}^{m} \hat{c}_i \; + \Phi_m - \Phi_0 \leq \; 3 \lg n + 1 + \Phi_m - \Phi_0.$$

Análise amortizada dos splay steps

Lembre-se que $\Phi_i = \sum_x r_i(x)$.

Ou seja, $\Phi_i \geq 0$ para todo i.

Então vale que o custo do SPLAY(x, S) é

$$\sum_{i=1}^{m} c_i \;=\; \sum_{i=1}^{m} \hat{c}_i \;+ \Phi_m - \Phi_0 \leq\; 3 \lg n + 1 + \Phi_m - \Phi_0.$$

É possível mostrar algo semelhante para inserções em splay trees.

Análise amortizada dos splay steps

Lembre-se que $\Phi_i = \sum_x r_i(x)$.

Ou seja, $\Phi_i \geq 0$ para todo i.

Então vale que o custo do SPLAY(x, S) é

$$\sum_{i=1}^{m} c_i = \sum_{i=1}^{m} \hat{c}_i + \Phi_m - \Phi_0 \leq 3 \lg n + 1 + \Phi_m - \Phi_0.$$

É possível mostrar algo semelhante para inserções em splay trees.

E disso é possível concluir que o custo amortizado por operação em uma splay tree é $\mathrm{O}(\lg n)$.

Inserções em splay trees

A inserção em splay trees é igual à inserção em ABB, seguida de uma chamada a SPLAY no elemento inserido.

Seja y_1 o pai de x após a inserção, e y_i o pai de y_{i-1} para $i = 1, \ldots, k$, onde y_k é a raiz da árvore.

Inserções em splay trees

A inserção em splay trees é igual à inserção em ABB, seguida de uma chamada a SPLAY no elemento inserido.

Seja y_1 o pai de x após a inserção, e y_i o pai de y_{i-1} para $i = 1, \ldots, k$, onde y_k é a raiz da árvore.

Considere a variação do potencial causado pela inserção.

r: potencial local antes da inserção
r': potencial local depois da inserção

Inserções em splay trees

A inserção em splay trees é igual à inserção em ABB, seguida de uma chamada a SPLAY no elemento inserido.

Seja y_1 o pai de x após a inserção, e y_i o pai de y_{i-1} para $i = 1, \ldots, k$, onde y_k é a raiz da árvore.

Considere a variação do potencial causado pela inserção.

r: potencial local antes da inserção
r': potencial local depois da inserção

$$
\begin{aligned}
\Delta\Phi &= \sum_{j=1}^{k} (r'(y_j) - r(y_j)) \\
&= \sum_{j=1}^{k} (\lg(s(y_j) + 1) - \lg(s(y_j)))
\end{aligned}
$$

Inserções em splay trees

A inserção em splay trees é igual à inserção em ABB, seguida de uma chamada a SPLAY no elemento inserido.

Seja y_1 o pai de x após a inserção, e y_i o pai de y_{i-1} para $i = 1, \ldots, k$, onde y_k é a raiz da árvore.

Considere a variação do potencial causado pela inserção.

r: potencial local antes da inserção
r': potencial local depois da inserção

$$
\begin{aligned}
\Delta\Phi &= \sum_{j=1}^{k}(r'(y_j) - r(y_j)) = \sum_{j=1}^{k}(\lg(s(y_j)+1) - \lg(s(y_j))) \\
&= \sum_{j=1}^{k} \lg \frac{s(y_j)+1}{s(y_j)}
\end{aligned}
$$

Inserções em splay trees

A inserção em splay trees é igual à inserção em ABB, seguida de uma chamada a SPLAY no elemento inserido.

Seja y_1 o pai de x após a inserção, e y_i o pai de y_{i-1} para $i = 1, \ldots, k$, onde y_k é a raiz da árvore.

Considere a variação do potencial causado pela inserção.

r: potencial local antes da inserção
r': potencial local depois da inserção

$$
\begin{aligned}
\Delta\Phi &= \sum_{j=1}^{k} (r'(y_j) - r(y_j)) = \sum_{j=1}^{k} (\lg(s(y_j)+1) - \lg(s(y_j))) \\
&= \sum_{j=1}^{k} \lg \frac{s(y_j)+1}{s(y_j)} = \lg\left(\prod_{j=1}^{k} \frac{s(y_j)+1}{s(y_j)}\right)
\end{aligned}
$$

Inserções em splay trees

Seja y_1 o pai de x após a inserção, e y_i o pai de y_{i-1} para $i = 1, \ldots, k$, onde y_k é a raiz da árvore.

Considere a variação do potencial causado pela inserção.

r: potencial local antes da inserção
r': potencial local depois da inserção

Inserções em splay trees

Seja y_1 o pai de x após a inserção, e y_i o pai de y_{i-1} para $i = 1, \ldots, k$, onde y_k é a raiz da árvore.

Considere a variação do potencial causado pela inserção.

r: potencial local antes da inserção
r': potencial local depois da inserção

$$\begin{aligned}
\Delta\Phi &= \sum_{j=1}^{k}(r'(y_j) - r(y_j)) = \sum_{j=1}^{k}(\lg(s(y_j)+1) - \lg(s(y_j))) \\
&= \sum_{j=1}^{k} \lg \frac{s(y_j)+1}{s(y_j)} = \lg\Big(\prod_{j=1}^{k} \frac{s(y_j)+1}{s(y_j)}\Big)
\end{aligned}$$

Inserções em splay trees

Seja y_1 o pai de x após a inserção, e y_i o pai de y_{i-1} para $i = 1, \ldots, k$, onde y_k é a raiz da árvore.

Considere a variação do potencial causado pela inserção.

r: potencial local antes da inserção
r': potencial local depois da inserção

$$\begin{aligned}
\Delta\Phi &= \sum_{j=1}^{k}(r'(y_j) - r(y_j)) = \sum_{j=1}^{k}(\lg(s(y_j)+1) - \lg(s(y_j))) \\
&= \sum_{j=1}^{k} \lg \frac{s(y_j)+1}{s(y_j)} = \lg\Big(\prod_{j=1}^{k} \frac{s(y_j)+1}{s(y_j)}\Big) \\
&= \lg\Big(\frac{s(y_2)}{s(y_1)} \cdot \frac{s(y_3)}{s(y_2)} \cdots \frac{s(y_k)}{s(y_{k-1})} \cdot \frac{s(y_k)+1}{s(y_k)}\Big)
\end{aligned}$$

(pois $s(y_j)+1 \leq s(y_{j+1}$

Inserções em splay trees

Seja y_1 o pai de x após a inserção, e y_i o pai de y_{i-1} para $i = 1, \ldots, k$, onde y_k é a raiz da árvore.

Considere a variação do potencial causado pela inserção.

r: potencial local antes da inserção
r': potencial local depois da inserção

$$\Delta\Phi = \sum_{j=1}^{k} \lg \frac{s(y_j)+1}{s(y_j)} = \lg\Big(\prod_{j=1}^{k} \frac{s(y_j)+1}{s(y_j)}\Big)$$

$$= \lg\Big(\frac{s(y_2)}{s(y_1)} \cdot \frac{s(y_3)}{s(y_2)} \cdots \frac{s(y_k)}{s(y_{k-1})} \cdot \frac{s(y_k)+1}{s(y_k)}\Big) \text{ (pois } s(y_j)+1 \leq s(y_{j+1}$$

$$= \lg\Big(\frac{s(y_k)+1}{s(y_1)}\Big)$$

Inserções em splay trees

Seja y_1 o pai de x após a inserção, e y_i o pai de y_{i-1} para $i = 1, \ldots, k$, onde y_k é a raiz da árvore.

Considere a variação do potencial causado pela inserção.

r: potencial local antes da inserção
r': potencial local depois da inserção

$$\begin{aligned}
\Delta\Phi &= \sum_{j=1}^{k} \lg \frac{s(y_j)+1}{s(y_j)} = \lg\Big(\prod_{j=1}^{k} \frac{s(y_j)+1}{s(y_j)}\Big) \\
&= \lg\Big(\frac{s(y_2)}{s(y_1)} \cdot \frac{s(y_3)}{s(y_2)} \cdots \frac{s(y_k)}{s(y_{k-1})} \cdot \frac{s(y_k)+1}{s(y_k)}\Big) \text{ (pois } s(y_j)+1 \leq s(y_{j+1} \\
&= \lg\Big(\frac{s(y_k)+1}{s(y_1)}\Big) \leq \lg n.
\end{aligned}$$

Inserções em splay trees

Seja y_1 o pai de x após a inserção, e y_i o pai de y_{i-1} para $i = 1, \ldots, k$, onde y_k é a raiz da árvore.

Considere a variação do potencial causado pela inserção.

r: potencial local antes da inserção
r': potencial local depois da inserção

$$\begin{aligned}
\Delta\Phi &= \sum_{j=1}^{k} \lg \frac{s(y_j)+1}{s(y_j)} = \lg \Big(\prod_{j=1}^{k} \frac{s(y_j)+1}{s(y_j)} \Big) \\
&= \lg \Big(\frac{s(y_2)}{s(y_1)} \cdot \frac{s(y_3)}{s(y_2)} \cdots \frac{s(y_k)}{s(y_{k-1})} \cdot \frac{s(y_k)+1}{s(y_k)} \Big) \quad \text{(pois } s(y_j)+1 \leq s(y_{j+1} \\
&= \lg \Big(\frac{s(y_k)+1}{s(y_1)} \Big) \leq \lg n.
\end{aligned}$$

Então $\hat{c} \leq 0 + \lg n = \lg n$. (inserção não faz rotações)

Concluindo a análise

Lembre-se que $\Phi_i \geq 0$ para todo i e agora $\Phi_0 = 0$.
(Começamos da árvore vazia.)

Concluindo a análise

Lembre-se que $\Phi_i \geq 0$ para todo i e agora $\Phi_0 = 0$.
(Começamos da árvore vazia.)

Então o custo total de m operações é

$$\begin{aligned}
\sum_{i=1}^{m} c_i &= \sum_{i=1}^{m} \hat{c}_i + \Phi_m - \Phi_0 \\
&\leq \sum_{i=1}^{m} (3 \lg n_i + 1) + \Phi_m - \Phi_0 \\
&\leq 3m \lg m + \lg m - 0 \\
&\leq (3m+1) \lg m.
\end{aligned}$$

Concluindo a análise

Lembre-se que $\Phi_i \geq 0$ para todo i e agora $\Phi_0 = 0$.
(Começamos da árvore vazia.)

Então o custo total de m operações é

$$\begin{aligned}
\sum_{i=1}^{m} c_i &= \sum_{i=1}^{m} \hat{c}_i + \Phi_m - \Phi_0 \\
&\leq \sum_{i=1}^{m} (3 \lg n_i + 1) + \Phi_m - \Phi_0 \\
&\leq 3m \lg m + \lg m - 0 \\
&\leq (3m + 1) \lg m .
\end{aligned}$$

Portanto o custo amortizado por operação é $\mathrm{O}(\lg m)$.

Concluindo a análise

Lembre-se que $\Phi_i \geq 0$ para todo i e agora $\Phi_0 = 0$.
(Começamos da árvore vazia.)

Então o custo total de m operações é

$$\begin{aligned}
\sum_{i=1}^{m} c_i &= \sum_{i=1}^{m} \hat{c}_i + \Phi_m - \Phi_0 \\
&\leq \sum_{i=1}^{m} (3 \lg n_i + 1) + \Phi_m - \Phi_0 \\
&\leq 3m \lg m + \lg m - 0 \\
&\leq (3m + 1) \lg m.
\end{aligned}$$

Portanto o custo amortizado por operação é $\mathrm{O}(\lg m)$.