

Heaps Binárias

Prof. Anderson Grandi Pires



Heap Binária: conceito

Tipo de Árvore Binária que possui duas propriedades:

a árvore é completa (as folhas no último nível estão situadas mais à esquerda da árvore)

todo nó na árvore deve seguir uma ordem específica, definindo assim uma *max-heap* ou *min-heap*



Heap Binária: conceito

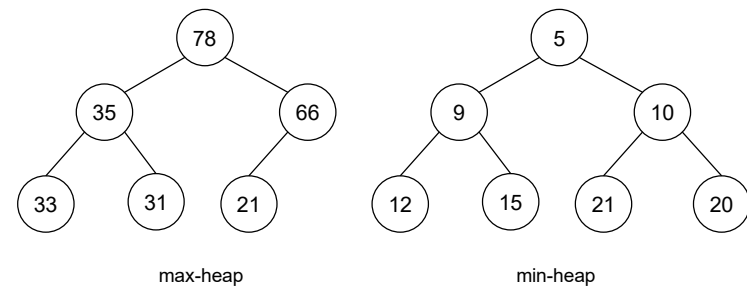
Tipo de Árvore Binária que possui duas propriedades:

max-heap: o conteúdo de um nó pai é maior ou igual ao conteúdo armazenado em cada um de seus filhos

min-heap: o conteúdo de um nó pai é menor ou igual ao conteúdo armazenado em cada um de seus filhos

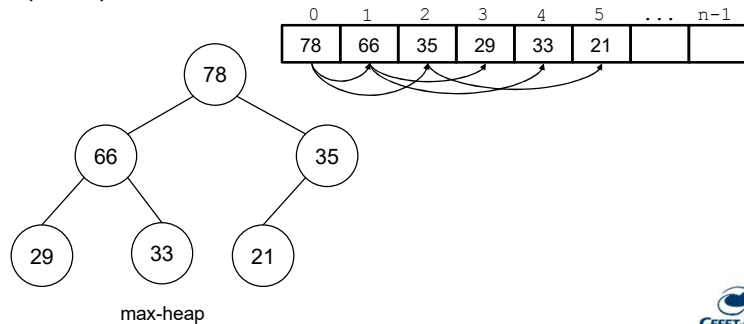


Heap Binária: tipos



Heap Binária: implementação

Pode ser eficientemente implementada usando um array (vetor)



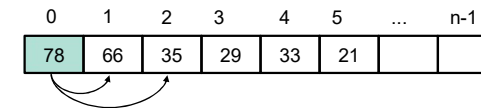
Heap Binária: implementação

Cálculo de parentesco

Pai: $\lfloor (i - 1)/2 \rfloor$, ou seja o piso de $(i - 1)/2$

Filho à esquerda: $2 \times i + 1$

Filho à direita: $2 \times i + 2$



Heap Binária: operações

Inserção

Um item é inserido no final do heap (próxima posição livre do vetor)

Caso o item inserido não conflita com as propriedades do heap, ele permanecerá onde foi inserido

Caso haja conflito, haverá necessidade de alterar a posição do item para atender às propriedades

Heap Binária: operações

Inserção em max-heap: passos

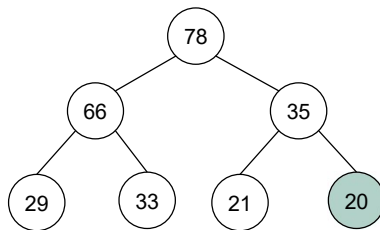
1. Insere o novo item no fim do heap
2. Compare o conteúdo do item com seu pai
 - i. Caso o item seja menor ou igual, pare
 - ii. Caso contrário, troque o item de posição com o pai
3. Volte ao item 2

Heap Binária: operações

Inserção em max-heap: exemplo

Inserir o valor 20

*Inserir o novo
item no fim
do heap*

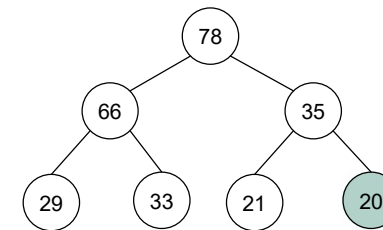


Heap Binária: operações

Inserção em max-heap: exemplo

Inserir o valor 20

*Verifique as
propriedades
do heap*

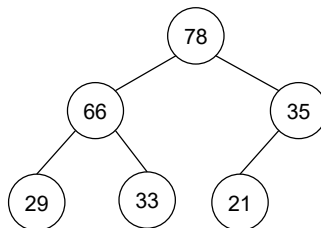


Heap Binária: operações

Inserção em max-heap: exemplo

Inserir o valor 99

*Inserir o novo
item no fim
do heap*

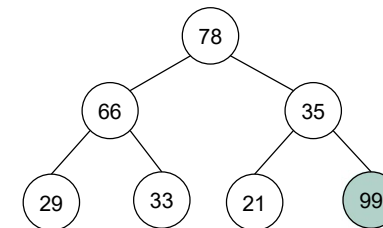


Heap Binária: operações

Inserção em max-heap: exemplo

Inserir o valor 99

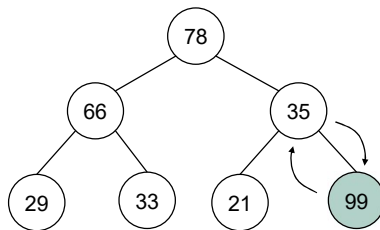
*Inserir o novo
item no fim
do heap*



Heap Binária: operações

Inserção em max-heap: exemplo

Inserir o valor 99



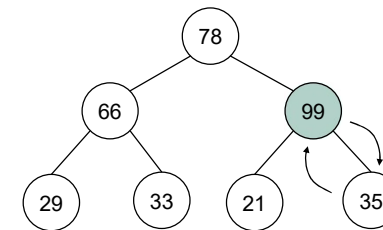
Verifique as
propriedades
do heap



Heap Binária: operações

Inserção em max-heap: exemplo

Inserir o valor 99



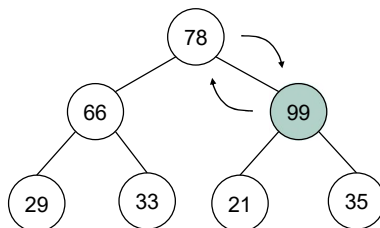
Verifique as
propriedades
do heap



Heap Binária: operações

Inserção em max-heap: exemplo

Inserir o valor 99



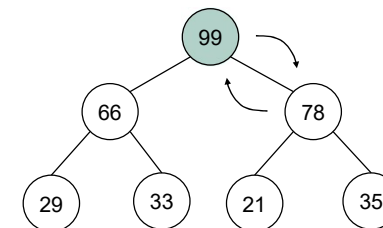
Verifique as
propriedades
do heap



Heap Binária: operações

Inserção em max-heap: exemplo

Inserir o valor 99



Verifique as
propriedades
do heap



Heap Binária: operações

Remoção

Em um heap, o primeiro item é sempre o maior (max-heap) ou o menor (min-heap)

Remover o máximo (max-heap) ou o mínimo (min-heap) significa remover o primeiro item

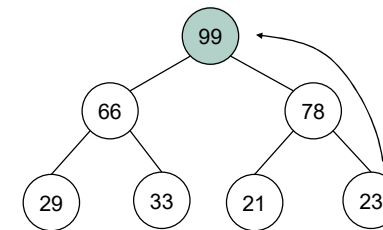
Para suprir a vacância gerada pela remoção, o último item do heap é inserido na primeira posição



Heap Binária: operações

Remoção em max-heap: exemplo

Remover o máximo



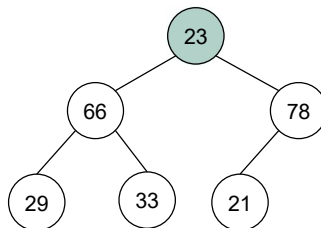
Último item
do heap é
armazenado
no lugar do
máximo



Heap Binária: operações

Remoção em max-heap: exemplo

Remover o máximo



Último item
do heap é
armazenado
no lugar do
máximo



Heap Binária: operações

Remoção

Para manter as **propriedades** do heap intactas, o “novo” primeiro item deverá ser “movimentado” para uma posição adequada

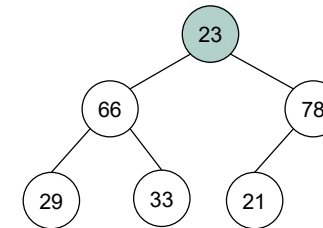


Heap Binária: operações*max-heapify***Movimentação do “novo” primeiro item (max-heap)**

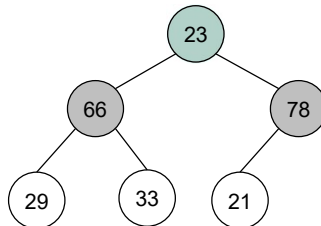
1. Verifique o “novo” item em relação aos seus filhos
Caso o “novo” item seja menor que algum de seus filhos, troque o item com o maior entre seus filhos
2. Repita o item 1 até que o “novo” item seja maior que seus filhos

**Heap Binária: operações****Remoção em max-heap: exemplo**

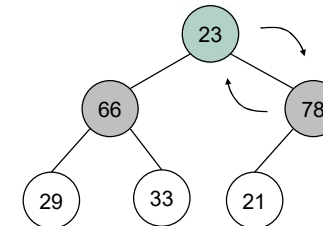
Remover o máximo

*Último item
do heap é
armazenado
no lugar do do
máximo***Heap Binária: operações****Remoção em max-heap: exemplo**

Remover o máximo

*Compare o
“novo” item
com seus
filhos***Heap Binária: operações****Remoção em max-heap: exemplo**

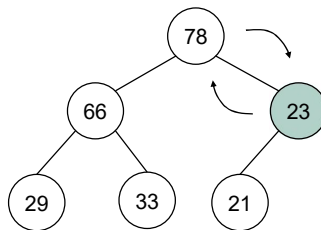
Remover o máximo

*Troque o item
com o maior
entre seus
filhos*

Heap Binária: operações

Remoção em max-heap: exemplo

Remover o máximo



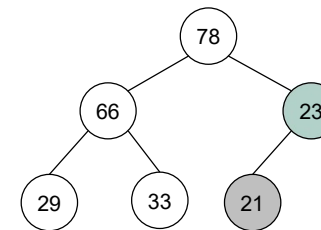
*Troque o item
com o maior
entre seus
filhos*



Heap Binária: operações

Remoção em max-heap: exemplo

Remover o máximo



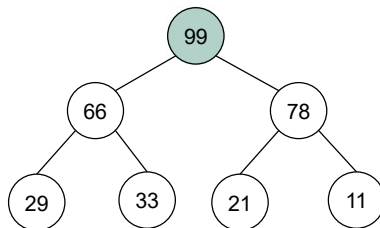
*Compare o
"novo" item
com seus
filhos*



Heap Binária: operações

Remoção em max-heap: outro exemplo

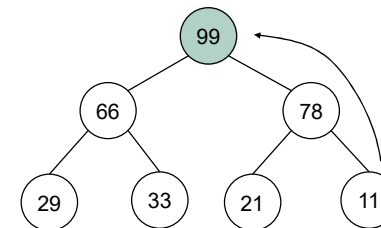
Remover o máximo



Heap Binária: operações

Remoção em max-heap: outro exemplo

Remover o máximo



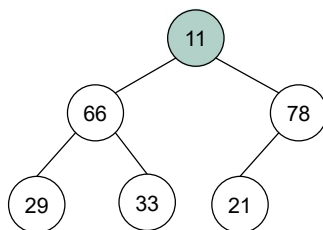
*Último item
do heap é
armazenado
no lugar do
máximo*



Heap Binária: operações

Remoção em max-heap: **outro exemplo**

Remover o máximo



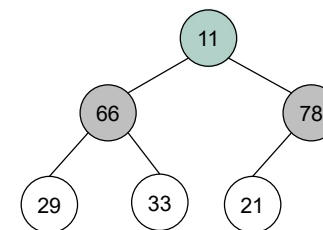
*Último item
do heap é
armazenado
no lugar do
máximo*



Heap Binária: operações

Remoção em max-heap: **outro exemplo**

Remover o máximo



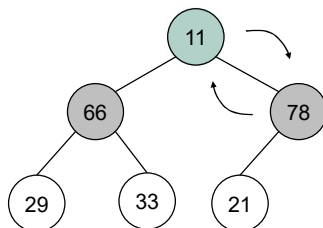
*Compare o
"novo" item
com seus
filhos*



Heap Binária: operações

Remoção em max-heap: **outro exemplo**

Remover o máximo



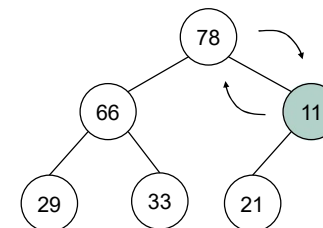
*Compare o
"novo" item
com seus
filhos*



Heap Binária: operações

Remoção em max-heap: **exemplo**

Remover o máximo



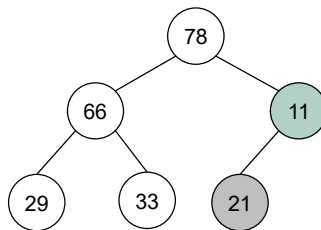
*Troque o item
com o maior
entre seus
filhos*



Heap Binária: operações

Remoção em max-heap: exemplo

Remover o máximo



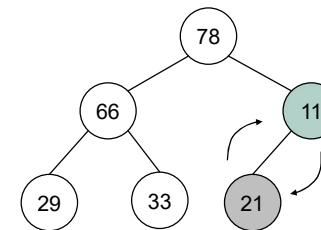
Compare o
"novo" item
com seus
filhos



Heap Binária: operações

Remoção em max-heap: exemplo

Remover o máximo

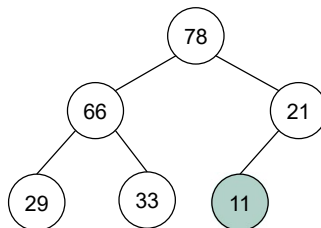


Troque o item
com o maior
entre seus
filhos

Heap Binária: operações

Remoção em max-heap: exemplo

Remover o máximo



Compare o
"novo" item
com seus
filhos



Heap Binária: operações

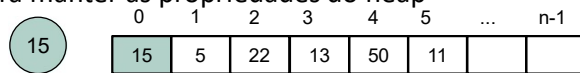
Construção de um max-heap a partir de um array

Método 1: executa-se a operação de inserção no heap descrita anteriormente para cada item do array

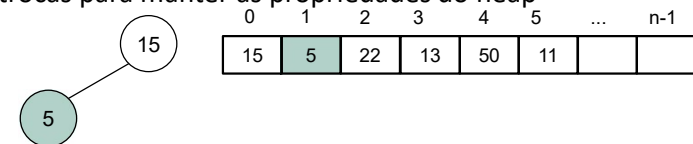
0	1	2	3	4	5	...	n-1
15	5	22	13	50	11		

Heap Binária: operações**Construção de um max-heap a partir de um array**

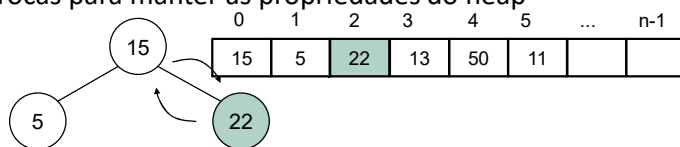
Método 1: executa-se a operação de inserção no heap descrita anteriormente para cada item do array e efetua as trocas para manter as propriedades do heap

**Heap Binária: operações****Construção de um max-heap a partir de um array**

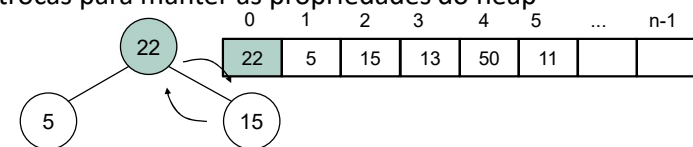
Método 1: executa-se a operação de inserção no heap descrita anteriormente para cada item do array e efetua as trocas para manter as propriedades do heap

**Heap Binária: operações****Construção de um max-heap a partir de um array**

Método 1: executa-se a operação de inserção no heap descrita anteriormente para cada item do array e efetua as trocas para manter as propriedades do heap

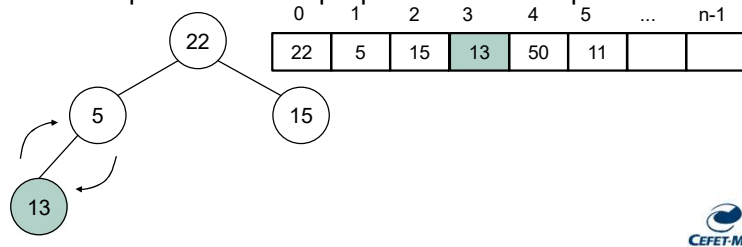
**Heap Binária: operações****Construção de um max-heap a partir de um array**

Método 1: executa-se a operação de inserção no heap descrita anteriormente para cada item do array e efetua as trocas para manter as propriedades do heap

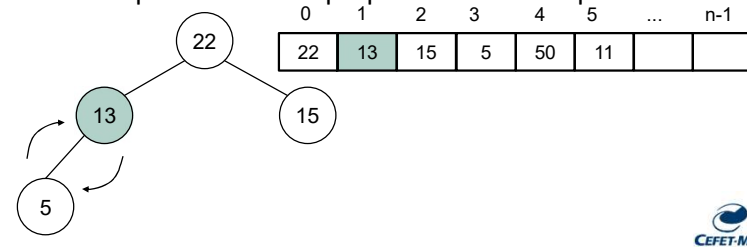


Heap Binária: operações**Construção de um max-heap a partir de um array**

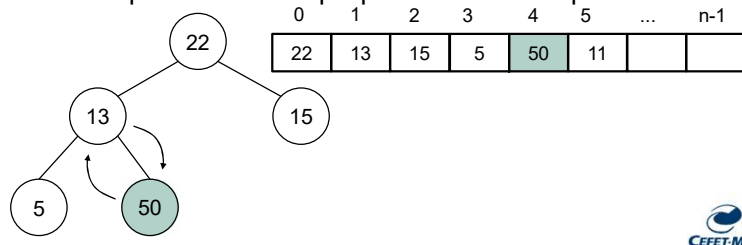
Método 1: executa-se a operação de inserção no heap descrita anteriormente para cada item do array e efetua as trocas para manter as propriedades do heap

**Heap Binária: operações****Construção de um max-heap a partir de um array**

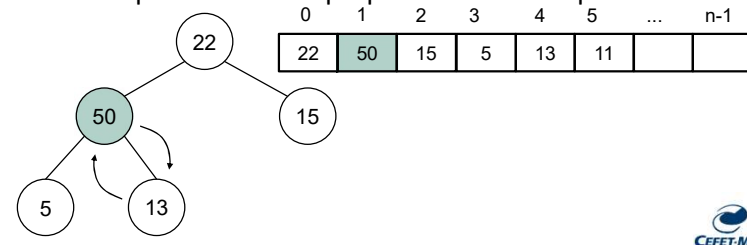
Método 1: executa-se a operação de inserção no heap descrita anteriormente para cada item do array e efetua as trocas para manter as propriedades do heap

**Heap Binária: operações****Construção de um max-heap a partir de um array**

Método 1: executa-se a operação de inserção no heap descrita anteriormente para cada item do array e efetua as trocas para manter as propriedades do heap

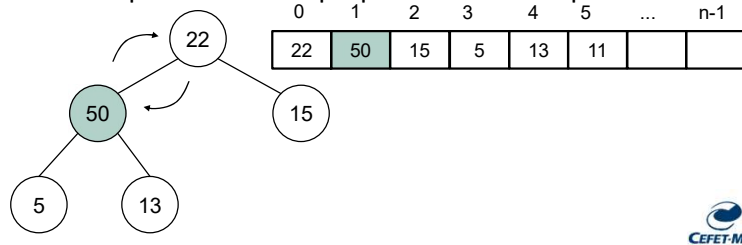
**Heap Binária: operações****Construção de um max-heap a partir de um array**

Método 1: executa-se a operação de inserção no heap descrita anteriormente para cada item do array e efetua as trocas para manter as propriedades do heap

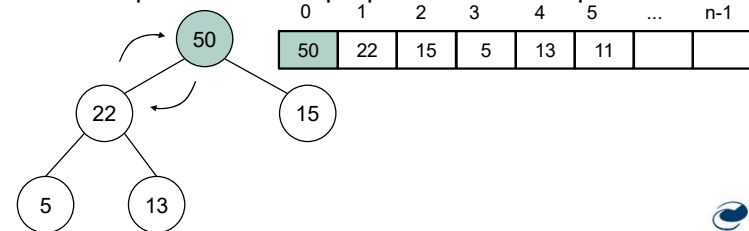


Heap Binária: operações**Construção de um max-heap a partir de um array**

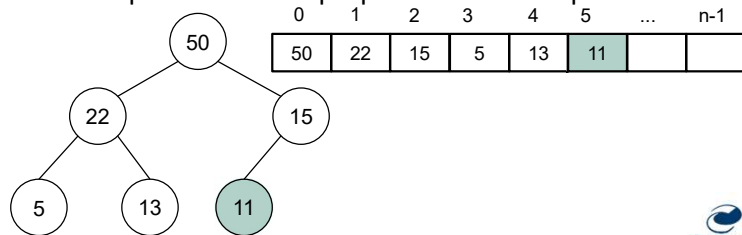
Método 1: executa-se a operação de inserção no heap descrita anteriormente para cada item do array e efetua as trocas para manter as propriedades do heap

**Heap Binária: operações****Construção de um max-heap a partir de um array**

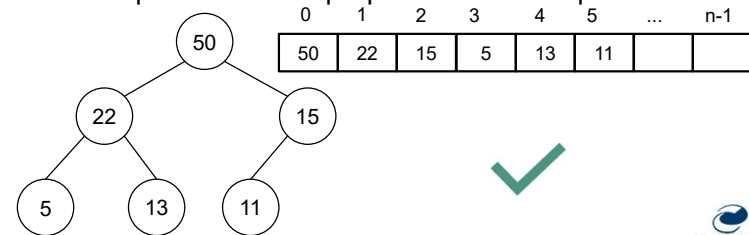
Método 1: executa-se a operação de inserção no heap descrita anteriormente para cada item do array e efetua as trocas para manter as propriedades do heap

**Heap Binária: operações****Construção de um max-heap a partir de um array**

Método 1: executa-se a operação de inserção no heap descrita anteriormente para cada item do array e efetua as trocas para manter as propriedades do heap

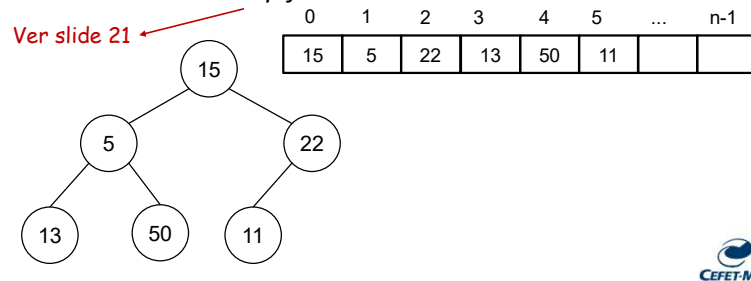
**Heap Binária: operações****Construção de um max-heap a partir de um array**

Método 1: executa-se a operação de inserção no heap descrita anteriormente para cada item do array e efetua as trocas para manter as propriedades do heap

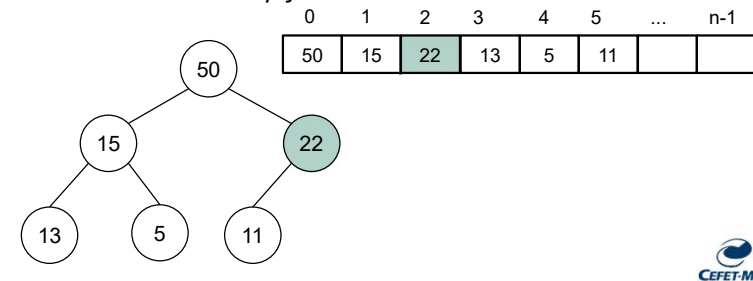


Heap Binária: operações

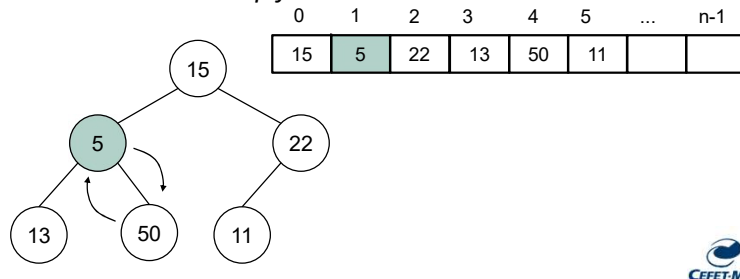
Construção de um max-heap a partir de um array

Método 2: insere todos os nós e depois os nós internos da árvore são *max-heapified* um de cada vez**Heap Binária: operações**

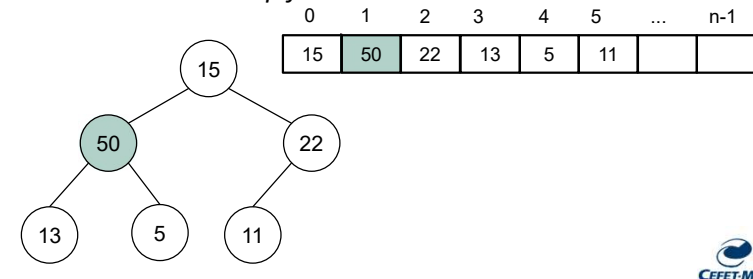
Construção de um max-heap a partir de um array

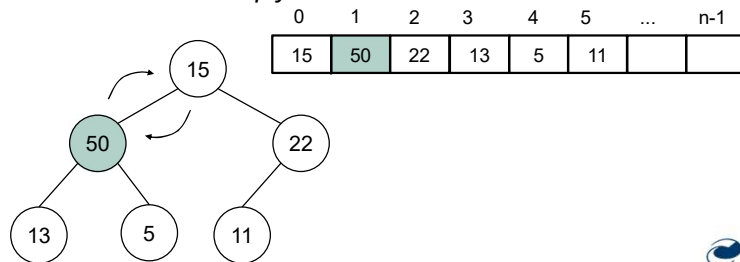
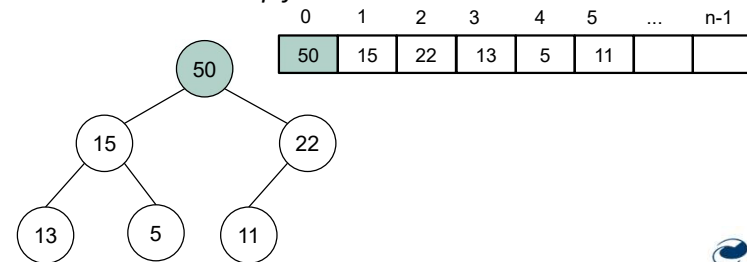
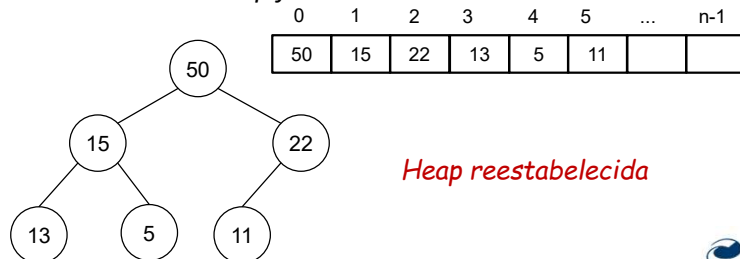
Método 2: insere todos os nós e depois os nós internos da árvore são *max-heapified* um de cada vez**Heap Binária: operações**

Construção de um max-heap a partir de um array

Método 2: insere todos os nós e depois os nós internos da árvore são *max-heapified* um de cada vez**Heap Binária: operações**

Construção de um max-heap a partir de um array

Método 2: insere todos os nós e depois os nós internos da árvore são *max-heapified* um de cada vez

Heap Binária: operações**Construção de um max-heap a partir de um array**Método 2: insere todos os nós e depois os nós internos da árvore são *max-heapified* um de cada vez**Heap Binária: operações****Construção de um max-heap a partir de um array**Método 2: insere todos os nós e depois os nós internos da árvore são *max-heapified* um de cada vez**Heap Binária: operações****Construção de um max-heap a partir de um array**Método 2: insere todos os nós e depois os nós internos da árvore são *max-heapified* um de cada vez**Algumas Referências**

- SZWARCFITER, Jayme Luiz; MARKENZON, Lilian. Estruturas de Dados e seus Algoritmos. 2ª Ed. Revisada. Rio de Janeiro: LTC, 1994.
- TENENBAUM, Aaron M.; LANGSAM, Yedidyah; AUGENSTEIN, Moshe J. Estruturas de Dados usando C. São Paulo: Pearson Makron Books, 1995.
- FEOFILOFF, Paulo. Algoritmos em linguagem C. Rio de Janeiro: Elsevier, 2009.
- UNIVESP TV. Engenharia da Computação: Estrutura de Dados. Disponível em: <https://bit.ly/32J9MwX>. Acesso em: 20/11/2020.
- Algorithm Tutor. Disponível em: <https://algorithmtutor.com>. Acesso em: 20/11/2020

