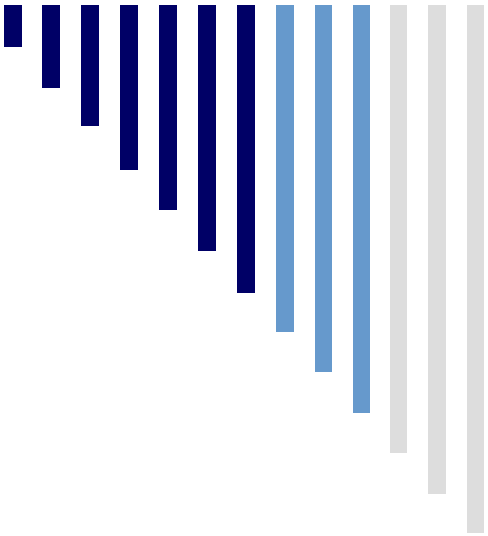


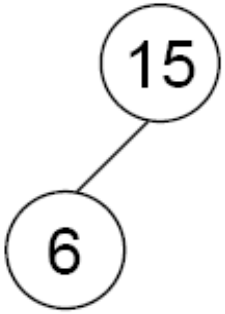
ÁRVORE HEAP

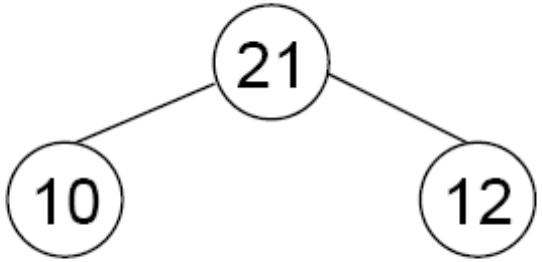
Conceito

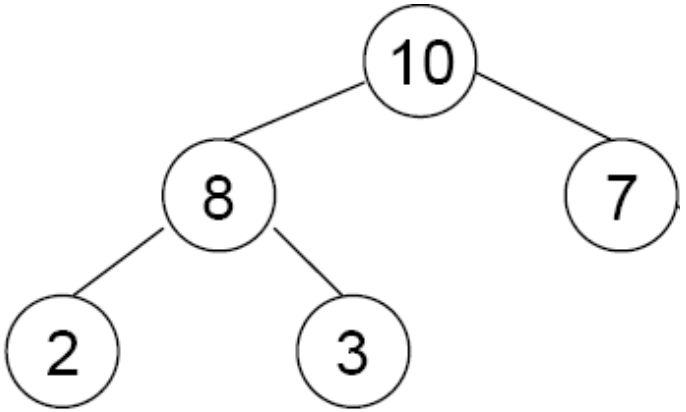
- É uma árvore binária com as seguintes propriedades:
  - O valor de cada nó não é menor do que os valores estocados em cada um de seus filhos;
  - A árvore é perfeitamente balanceada e as folhas no último nível estão todas nas posições mais à esquerda.

Exemplos

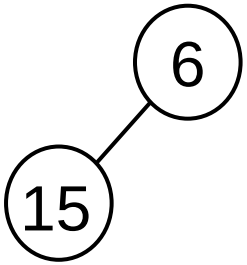
HEAP MÁXIMA:

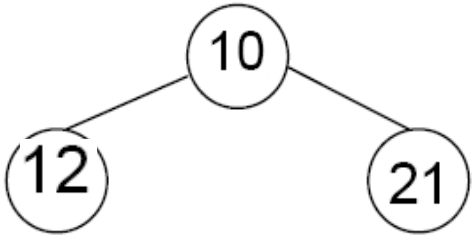






HEAP MÍNIMA:





Aplicação

- Fila de prioridades
  - Estrutura de dados que suporte as operações de retirar o item de maior ou menor valor.
  - Exemplo:
    - SO – filas de eventos que devem ocorrer.
    - Sistemas de gerência de memória usam a técnica de substituir a página menos utilizada na memória principal por uma nova página.
- HeapSort:
  - Algoritmo de ordenação que utiliza a estrutura heap.

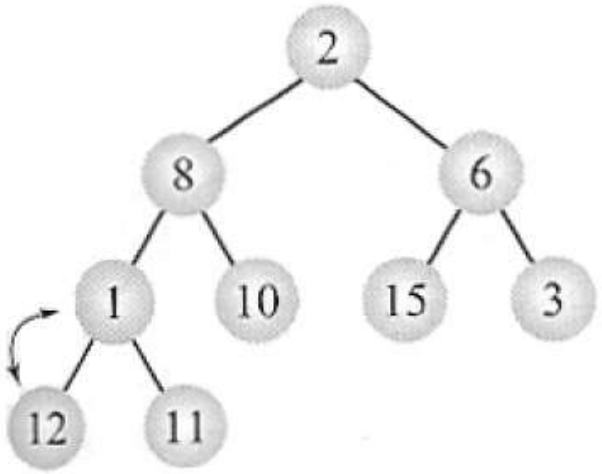
Estrutura

- Um heap é definido como uma sequência de itens com chaves
  - c[1], c[2], ... , c[n]
- Tal que
  - C[i] ≥ c[2i]
  - C[i] ≥ c[2i + 1]
- Para todo i = 1, 2, ... n/2

O Nó i é pai dos nós 2i e 2i + 1.

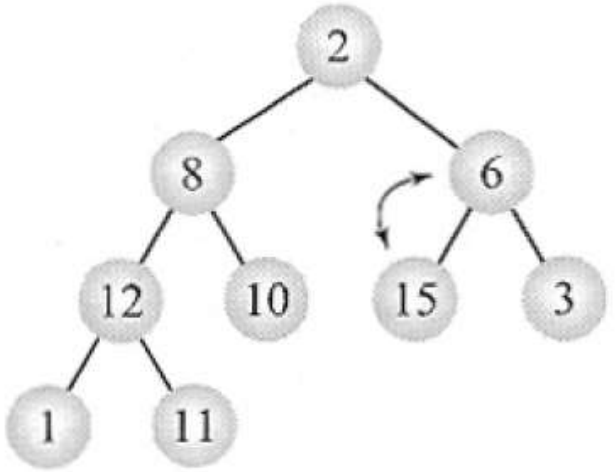
Geralmente é implementada com vetor!

Transformando um vetor numa árvore heap



2	8	6	1	10	15	3	12	11
1	2	3	4	5	6	7	8	9

(a)

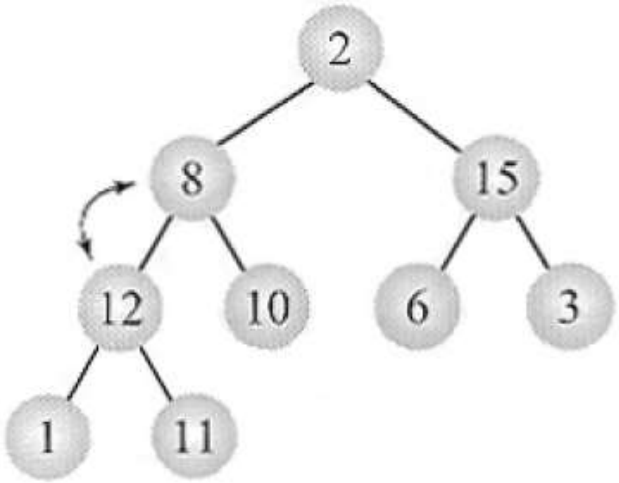


2	8	6	12	10	15	3	1	11

(b)

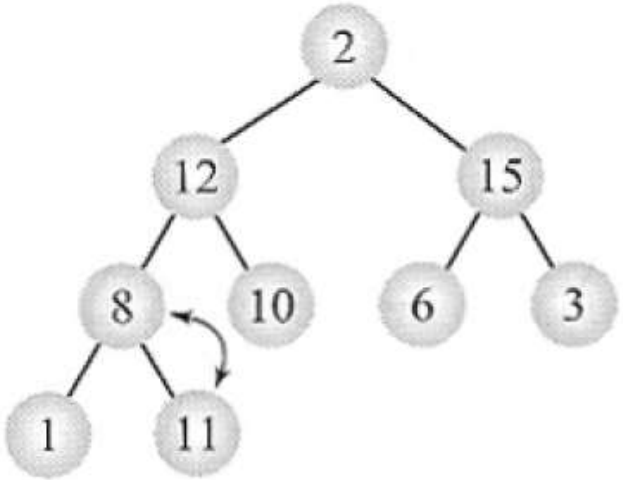
i
2*i
2*i + 1

Transformando um vetor numa árvore heap



2	8	15	12	10	6	3	1	11

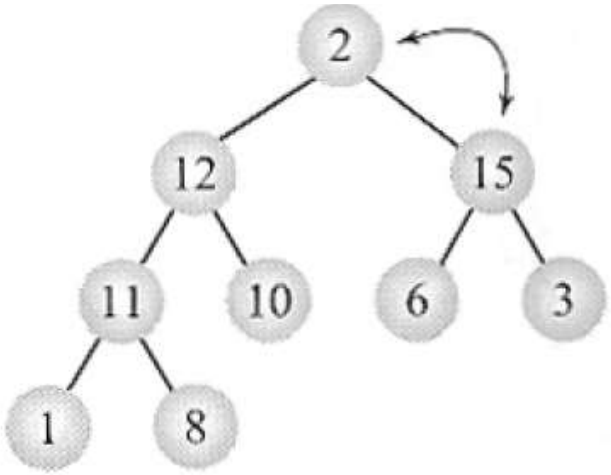
(c)



2	12	15	8	10	6	3	1	11

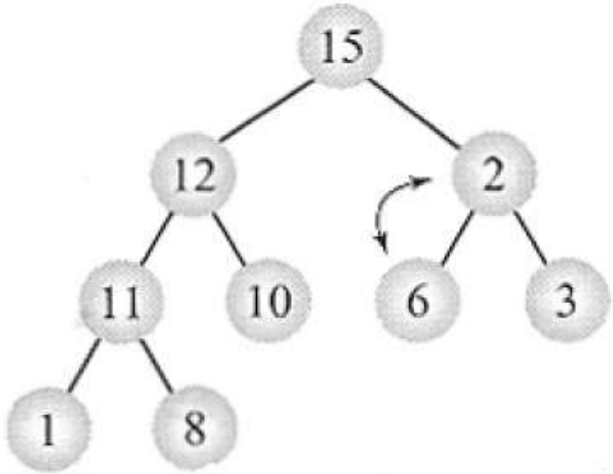
(d)

Transformando um vetor numa árvore heap



2	12	15	11	10	6	3	1	8

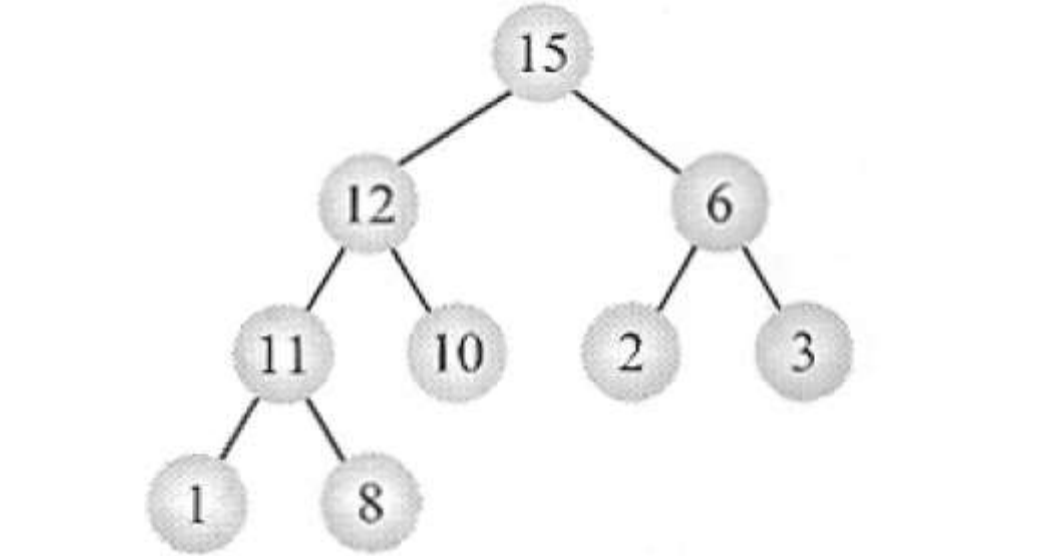
(e)



15	12	2	11	10	6	3	1	8

(e)

Transformando um vetor numa árvore heap



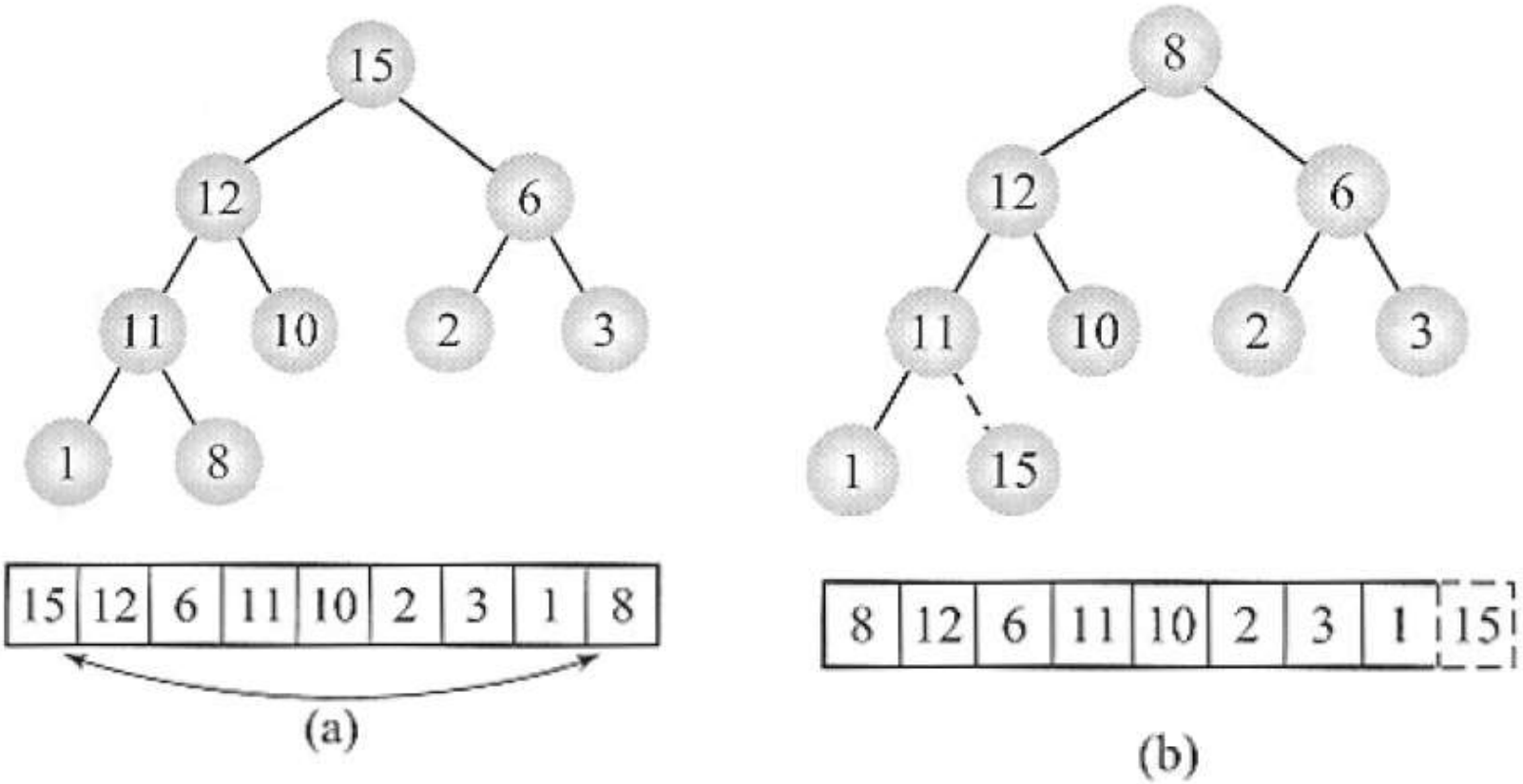
(f)

HeapSort

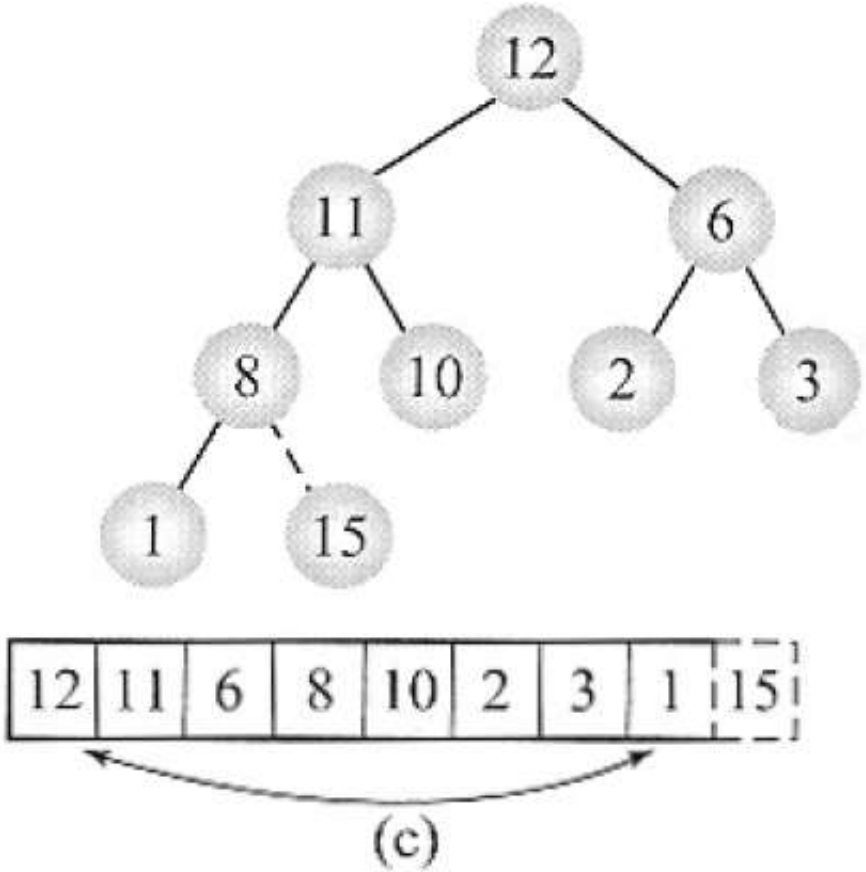
- Possui o mesmo princípio de funcionamento da ordenação por seleção.

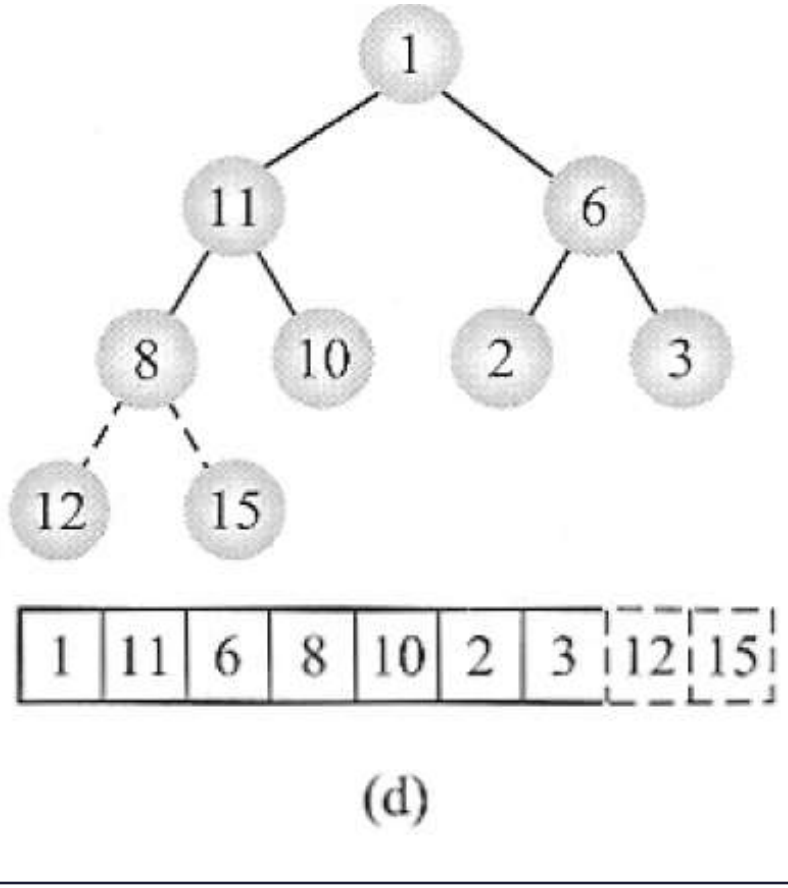
- Coloca o maior elemento (raiz) no final do vetor e regenera a estrutura heap para os n-1 elementos.
  - Repita esse processo para os n-1 itens restantes, depois com os n – 2, até que reste apensa 1 item.

Execução do Heapsort (1)

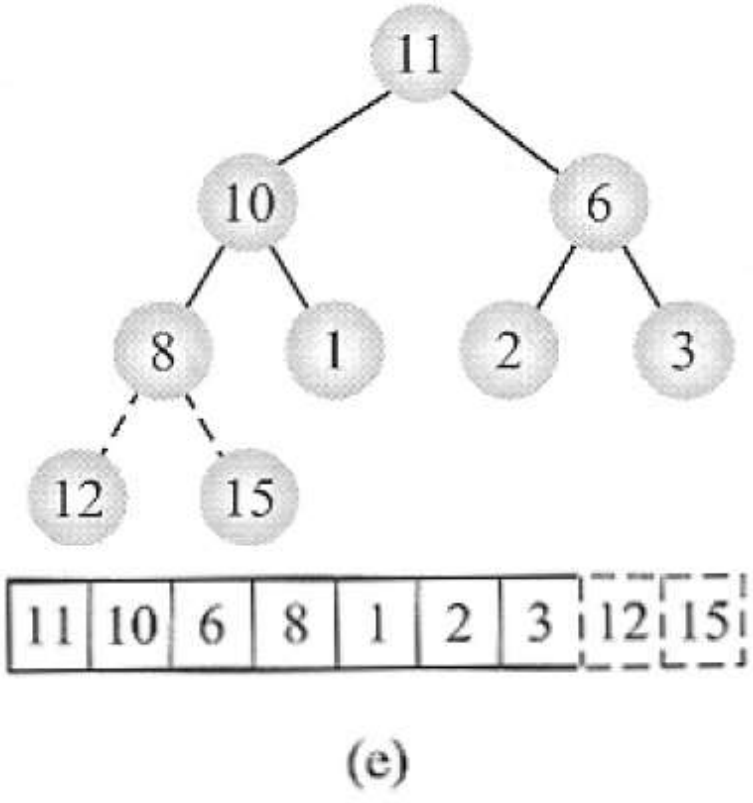


Execução do Heapsort (2)

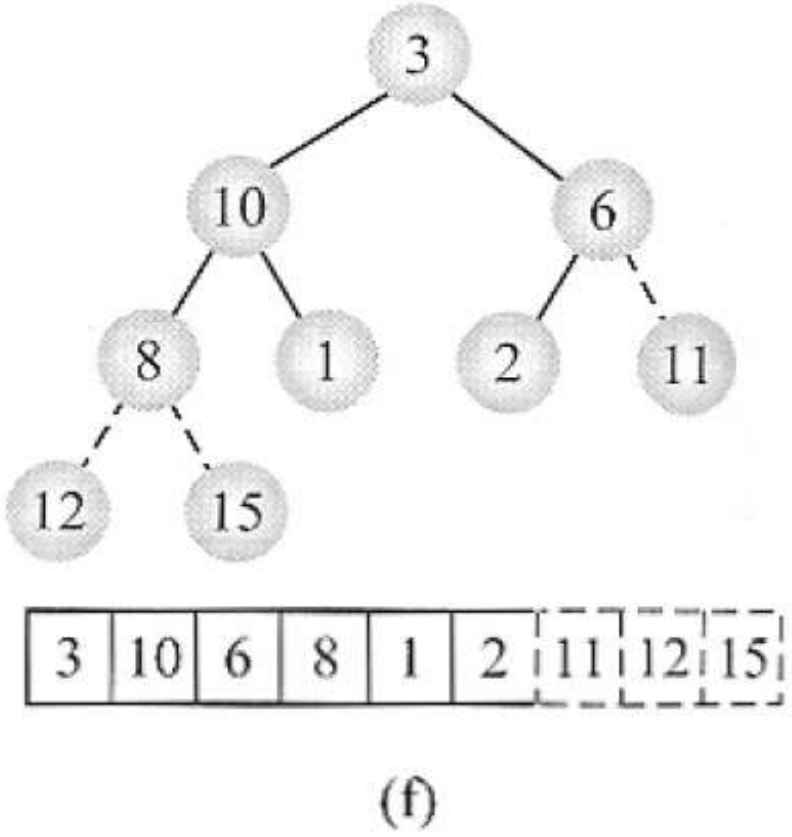




Execução do Heapsort (3)

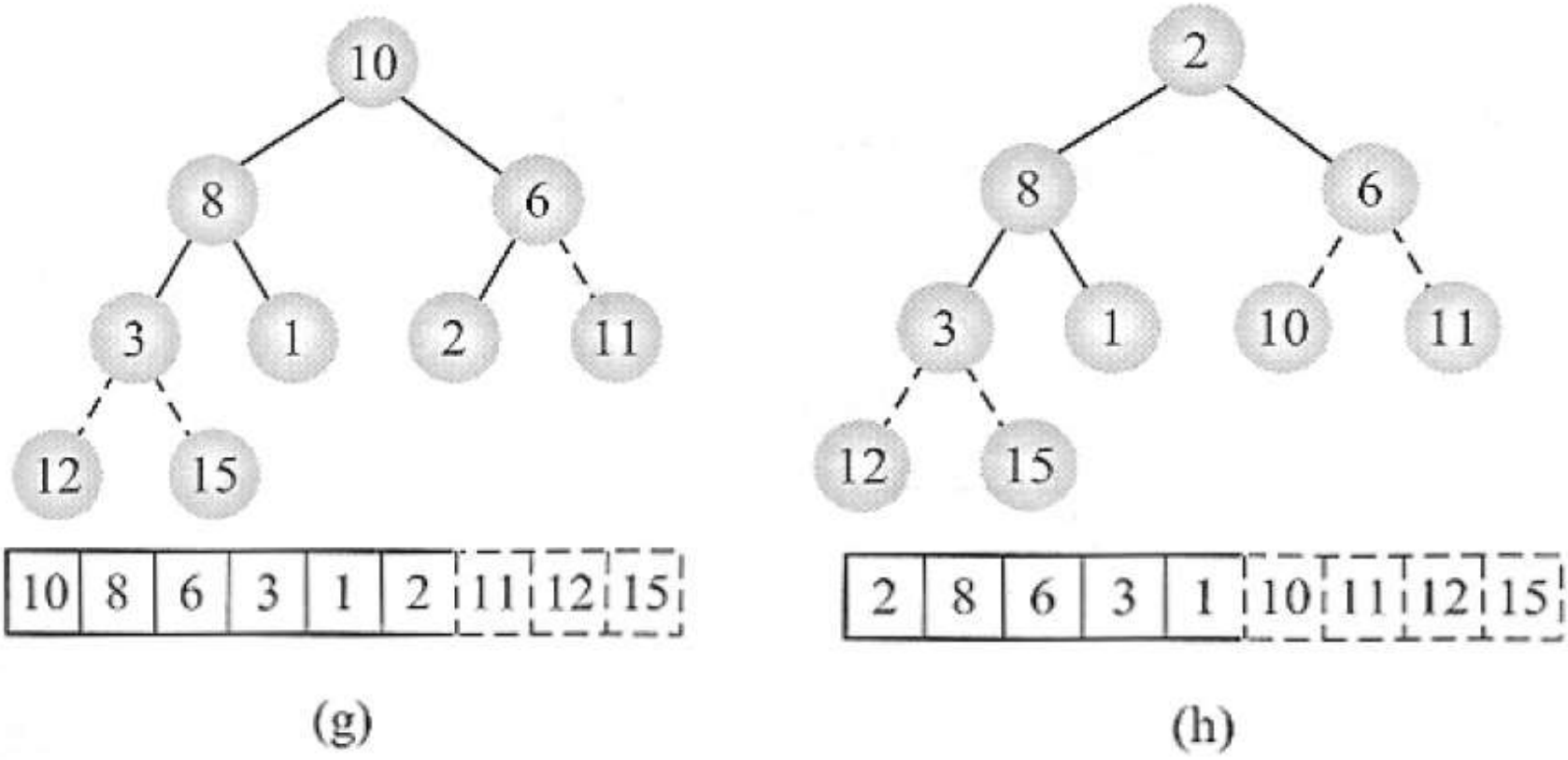


(e)

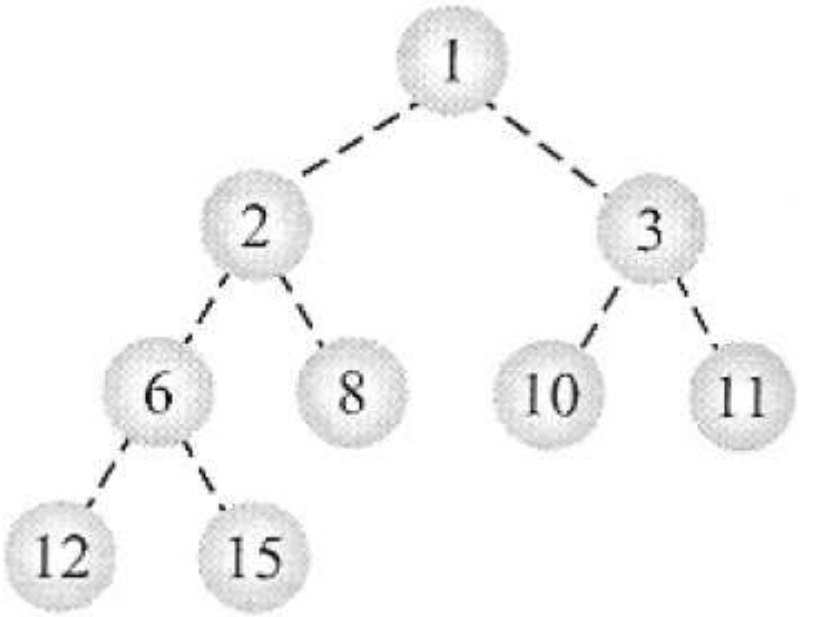


(f)

Execução do Heapsort (4)

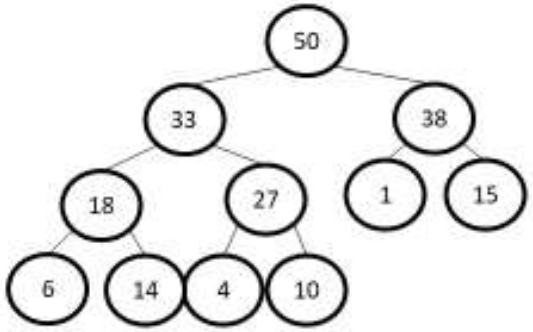


Execução do Heapsort (5)

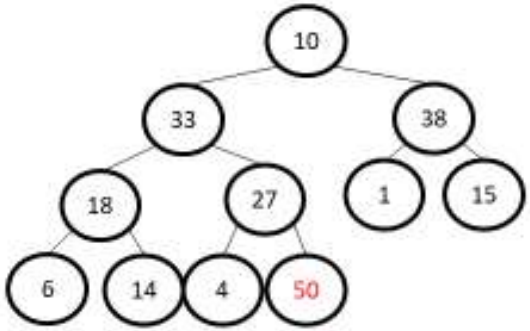


(i)

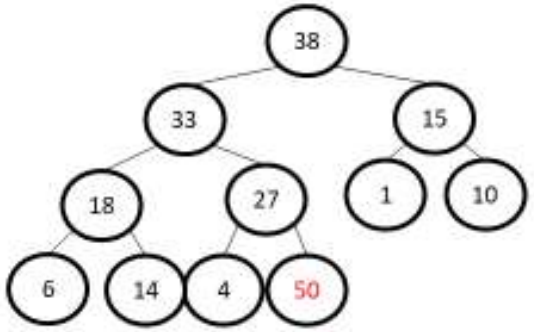
Outro exemplo (1)



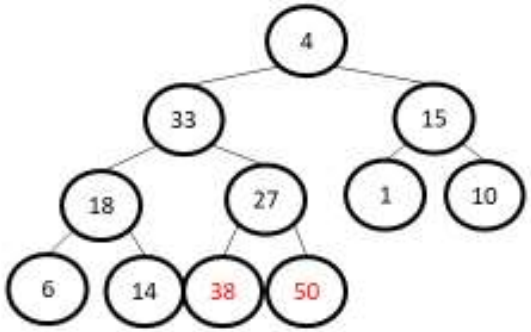
50 33 38 18 27 1 15 6 14 4 10



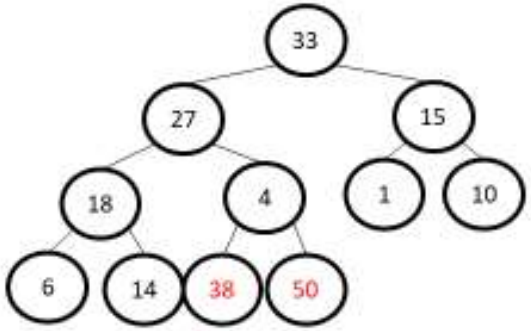
10 33 38 18 27 1 15 6 14 4 50



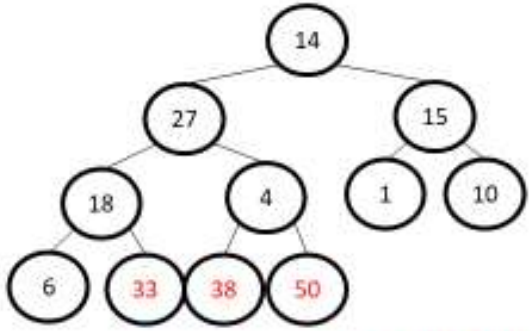
38 33 15 18 27 1 10 6 14 4 50



4 33 15 18 27 1 10 6 14 38 50

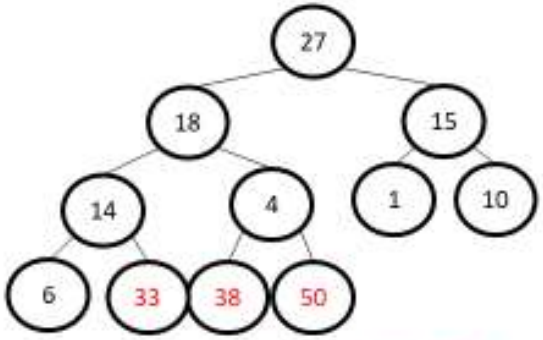


33 27 15 18 4 1 10 6 14 38 50

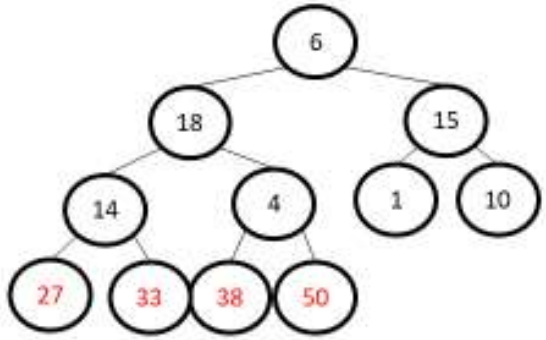


14 27 15 18 4 1 10 6 33 38 50

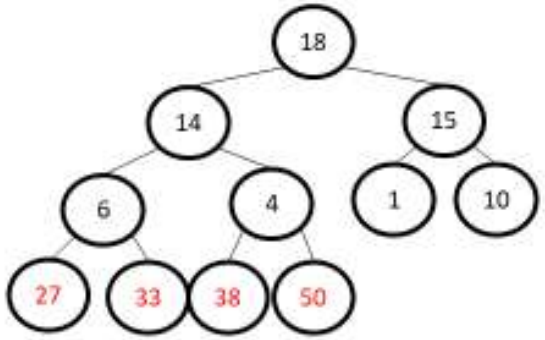
Outro exemplo (2)



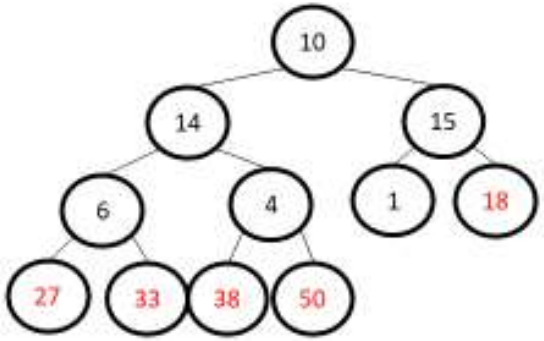
27 18 15 14 4 1 10 6 33 38 50



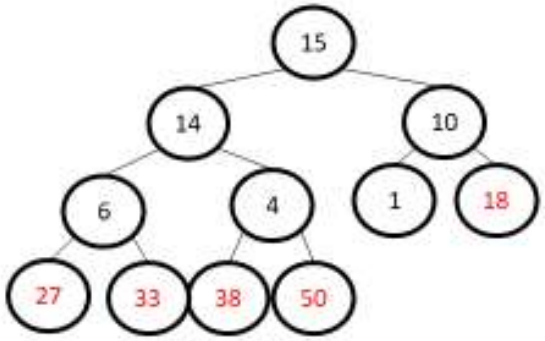
6 18 15 14 4 1 10 27 33 38 50



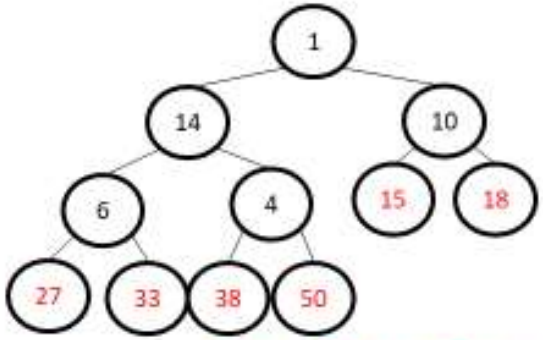
18 14 15 6 4 1 10 27 33 38 50



10 14 15 6 4 1 18 27 33 38 50

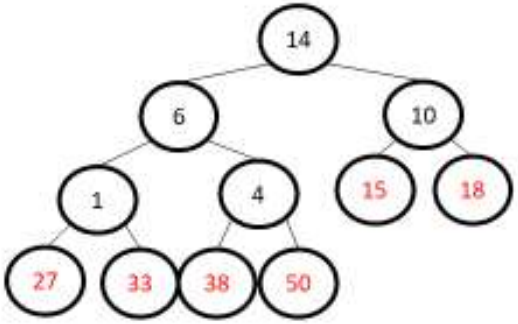


15 14 10 6 4 1 18 27 33 38 50

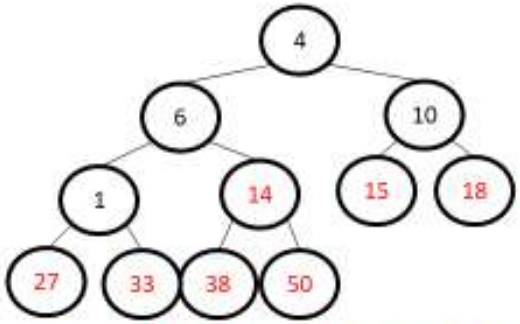


1 14 10 6 4 15 18 27 33 38 50

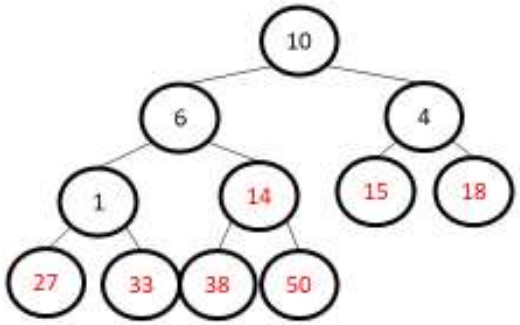
Outro exemplo (3)



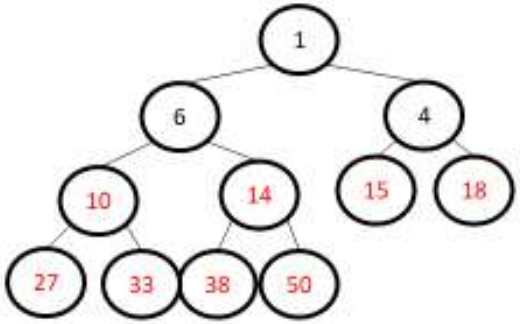
14 6 10 1 4 15 18 27 33 38 50



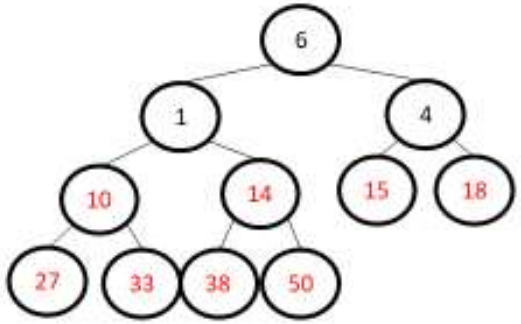
4 6 10 1 14 15 18 27 33 38 50



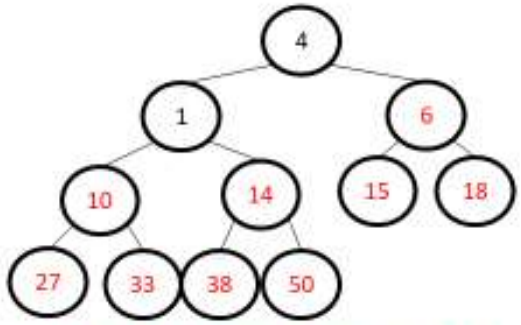
10 6 4 1 14 15 18 27 33 38 50



1 6 4 10 14 15 18 27 33 38 50

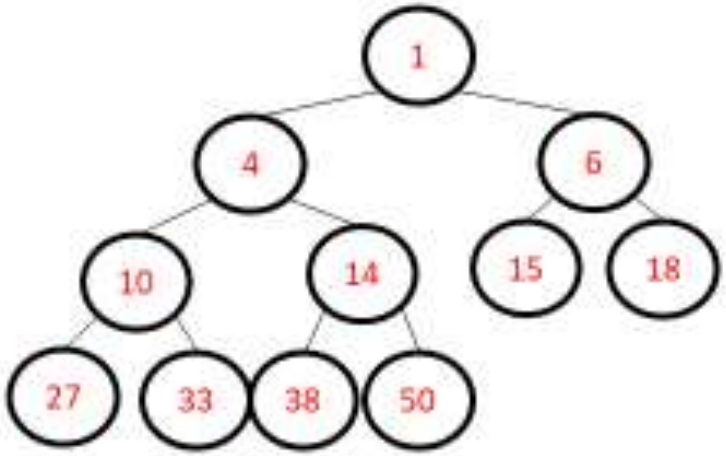


6 1 4 10 14 15 18 27 33 38 50



4 1 6 10 14 15 18 27 33 38 50

Outro exemplo (4)



Link para o vídeo da aula

- https://drive.google.com/file/d/1ehNrGCo2FyO47pHc7tkbqZaZulnnFYUo/view?usp=sharing