

## Explicação da Aula 5

Segunda-Feira 23/09/2019

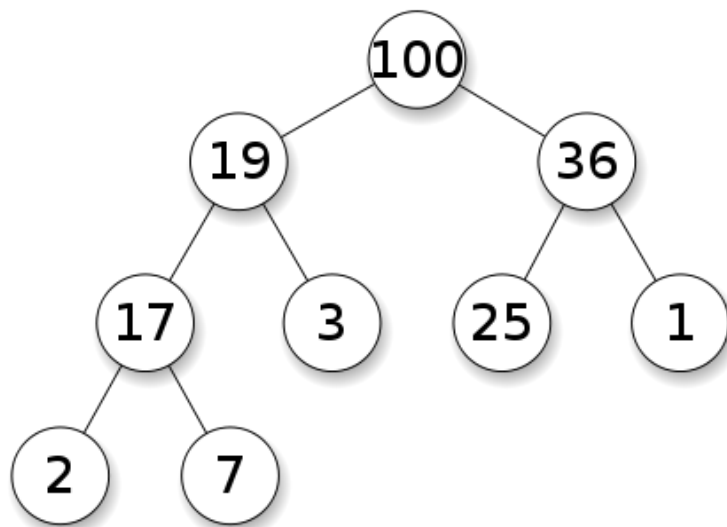
Heap-Sort:

- Algoritmo de ordenação, seu tempo de ordenação é  $O(n \lg n)$
- Ordena no local
- Usa a estrutura de dados *heap*

*Estrutura de dados heap* é tipo específico de uma árvore binária. Uma estrutura de prioridades, na forma de **árvore binária completa**, que representa uma ordem parcial entre os elementos do conjunto. Características:

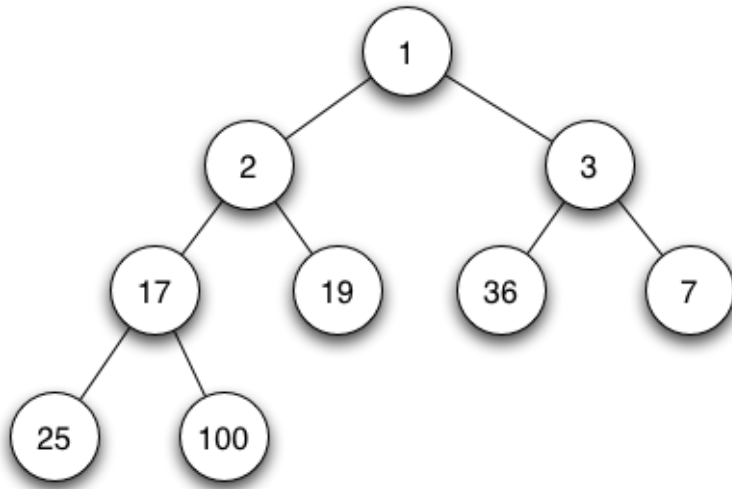
1. A árvore está completamente preenchida em todos os níveis exceto talvez o mais baixo;
2. Nível mais baixo preenchido a partir da esquerda

Tipos:



- Heap Máximo:

U+2713 Para todo nó o **pai é maior ou igual** aos filhos:



- Heap Mínimo:
- Curiosidade:
- um heap que armazena  $n$  keys tem altura  $O(\log n)$   $n = t_n$
- o numero total de nós em  $(t_n)$  em uma árvore binária cheia de altura  $h$  é:  
 $t_n = (2^h - 1)$
- formula, numa arvore binária completa, a altura  $h$  de uma árvore é dada por:  $h = \log_2(t_n + 1)$

#### MAX-HEAPIFY

```

Max-Heapify(A, i)
1. l = Left(i)
2. r = Right(i)
3. if l <= A.tamanho_do_heap e A[l] > A[i]
4.   maior = l
5. else maior = i
6. if r <= A.tamanho_do_heap e A[r] > A[maior]
7.   maior = r
8. if maior != i
9.   troca A[i] com A[maior]
10.  Max-Heapify(A, maior)
  
```