made for free at coggle.it

Para garantir que a heap esteja cumprindo as condições são realizadas comparações para trocar as chaves de cada nó se necessário

Todos os elementos são conhecidos antes da criação da heap, cada elemento é inserido de

forma sequencial

Os elementos não são conhecidos e são inseridos um por um, e o processo de heapify

(comparações sucessivas para garantir o

cumprimento das condições) ocorre a cada inserção

Uma priority queue pode ser definida por um multiset de itens ordenavéis por uma característica - prioridade É uma estrutura de dados parcialmente ordenada utilizada para implementar uma priority queue (fila de prioridade) Uma estrutura de heap pode ser ilustrada e Bottom up construction Heap definida como uma árvore binária, seguindo as seguintes condições: Topdown construction List (sorted/unsorted) Implementações BST Heap e Heapsort AVL Heapsort Algoritmo de ordenação com heap Primeiro fazemos a criação da heap com o método bottom up em cima de um array e depois fazemos a operação de delete no índice n-1

A árvore deve ser parcialmente completa, apenas as folhas direitas do último nível da árvore podem ser vazias

Shape Property

Parental Dominance

Cada nó da heap deve obedecer o seguinte critério: a chave de um nó deve ser >= ou <= (maxheap, minheap) do que as chaves dos seus filhos