



UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO

COMP0405 - ESTRUTURA DE DADOS

GRUPO B

Willian Mota Oliveira - 201800017520

Yohan Alexander Dantas de França - 201800017208

Relatório do Projeto Heap

São Cristóvão - SE

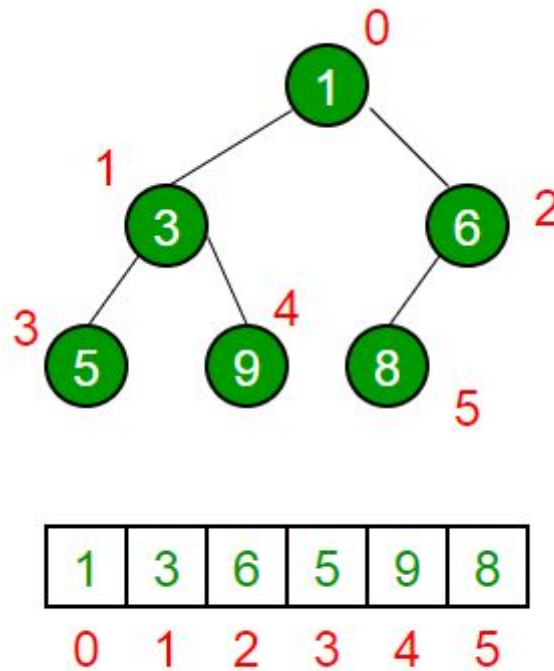
Fevereiro de 2020

SUMÁRIO

- 1. Introdução**
- 2. Descrição**
- 3. Aplicação**
- 4. Referências bibliográficas**
- 5. Implementação em C**

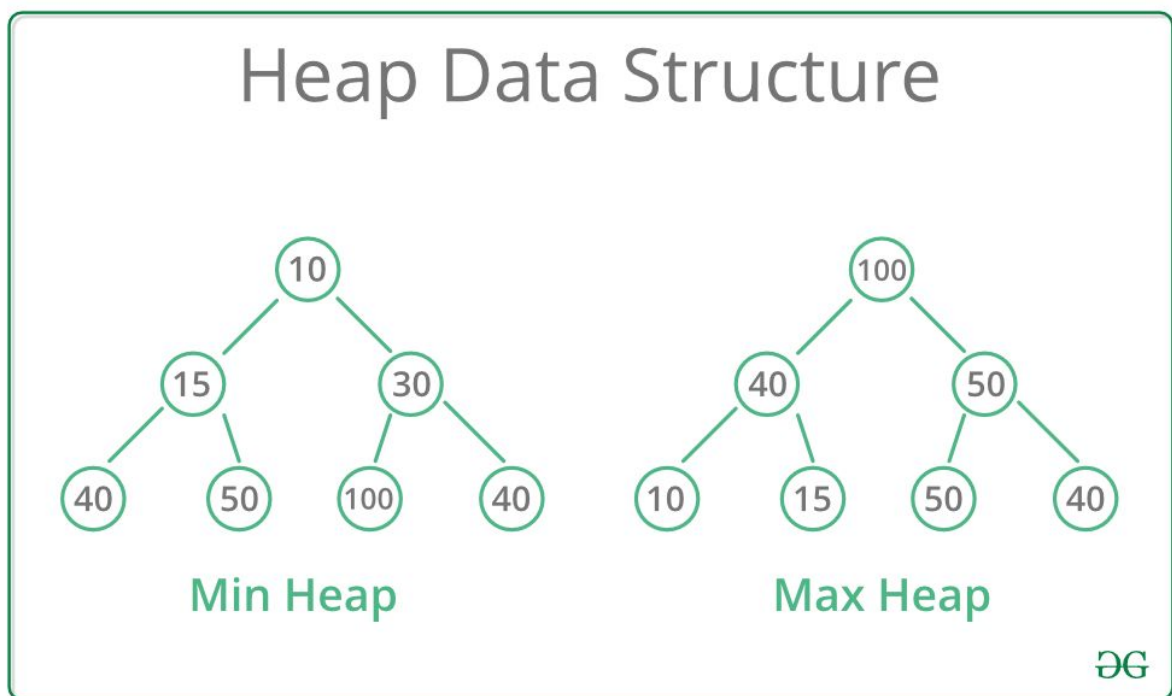
1. INTRODUÇÃO

Um heap é uma estrutura de dados baseada em árvore, ou seja, organiza os dados de forma hierárquica, na qual a árvore é uma árvore binária completa. A estrutura hierárquica dos heaps geralmente é implementada de forma implícita em um array.



Geralmente, os HEAPs podem ser de dois tipos:

- Heap Máximo: em um heap máximo, a chave presente no nó raiz deve ser maior entre as chaves presentes em todos os filhos. A mesma propriedade deve ser verdadeira recursivamente para todas as subárvores nessa árvore binária.
- Heap Mínimo: em um heap mínimo, a chave presente no nó raiz deve ser mínima entre as chaves presentes em todos os filhos. A mesma propriedade deve ser verdadeira recursivamente para todas as subárvores nessa árvore binária.



2. DESCRIÇÃO

A árvore binária do heap deve estar completa até pelo menos seu penúltimo nível e, se o seu último nível não estiver completo, todos os nós do último nível deverão estar agrupados à esquerda. Assim, as inserções em uma heap são feitas sequencialmente no array em que esta é implementada, com certas manipulações para a manutenção de suas propriedades, além disto as inserções são sempre feitas nas folhas, mais precisamente na folha livre mais à esquerda, essa inserção pode não respeitar as invariantes da heap, por isso é necessário um processo chamado heapify após a inserção para garantir que as invariantes sejam mantidas.

Considerando n como o número de elementos de um heap, inserção e remoção têm complexidade da ordem $O(\log(n))$.

3. APLICAÇÃO

O heap é uma implementação maximamente eficiente de um tipo de dados abstrato chamado de fila de prioridade e, de fato, as filas de prioridade são geralmente chamadas de "heaps", independentemente de como elas podem ser implementadas. Em uma heap, o elemento de prioridade mais alta (ou mais baixa) é sempre armazenado na raiz. No entanto, uma heap não é uma estrutura classificada, ela pode ser considerada parcialmente ordenada. Uma heap é uma estrutura de dados útil quando é necessário remover repetidamente o objeto com a prioridade mais alta (ou mais baixa).

Por isso, são usadas como uma estrutura de dados para o algoritmo de classificação heapsort. Heaps também são cruciais em vários algoritmos de grafo eficientes, como o algoritmo de Dijkstra. Quando uma heap é uma árvore binária completa, ela possui a menor altura possível.

4. REFERÊNCIAS BIBLIOGRÁFICAS

- <https://pt.wikipedia.org/wiki/Heap>