

ESTRUTURA HEAP – MATHEUS HENRIQUE BUTKOSKI SILVA

Heap é uma estrutura de dados que pode ser visualizada como uma árvore binária quase completa em que cada nó da árvore é ocupado por um elemento do **vetor** e possui as seguintes propriedades:

-A árvore é completa até o penúltimo nível

-No último nível as folhas estão o mais à esquerda possível

-O conteúdo de um nó é maior ou igual ao conteúdo dos nós na subárvore enraizada nele(max-heap)

A primeira posição disponível em um heap representará sua raiz que, em que o filho esquerdo será igual a $2i$ e o filho direito será $2i+1$.

Dessa forma, o vetor do heap não possuirá espaços em branco e um nó sempre ocupará uma posição.

Para adicionar um elemento no Heap basta inseri-lo na ultima posição do vetor n . Como descrito na terceira propriedade, o conteúdo de um nó deve ser maior ou igual aos da subárvore. Dessa forma, será preciso comparar o valor com os nós pais desse elemento, e seguir trocando-os de posição até encontrar uma posição que satisfaz a propriedade.

Já na remoção, sempre será necessário remover o nó de maior prioridade, ou seja, a raiz. Para isso, deve-se retirar a raiz da posição e no lugar dele inserir o último do heap, que estará à esquerda. Após isso, deve-se comparar a nova raiz com os dois filhos e troca-la de posição sempre com o maior filho, até encontrar o local em que não haverá filhos maiores que o nó, e assim o heap estará estabelecido novamente.

Para realizar a busca, pode-se utilizar os métodos padrões de busca em um vetor caso a função vise apenas buscar os elementos em um heap já pronto. Cria-se uma propriedade for , que repetirá a busca por todos os elementos no heap, e para saber sua localização exata e altura na árvore, utiliza-se $(i-1) / 2$, que sempre representará a posição do pai do elemento buscado, e seus filhos esquerdos e direitos respectivamente, $2i$ e $2i+1$.

Uma heap baseia-se na estrutura de uma fila de prioridade, diferente de uma árvore binária, em que o maior valor se tornará filho direito e o menor filho esquerdo, a heap permite que os valores fiquem tanto na direita quanto na esquerda, a única condição deve ser que os nós filhos devem ser menores que os nós pais. Dessa forma, se torna simples formar uma heap balanceada logo na inserção, pois caso o valor não encontre sua prioridade na posição inicial, basta percorrer a heap até encontrar a posição correspondente, lembrando sempre de deixar as folhas do último nível na esquerda, caso ele não esteja completo.

Referências:

https://www.youtube.com/watch?v=v3xF3bKChnI&ab_channel=CarlaQuemDisse

https://www.youtube.com/watch?v=hyedRznKnjQ&t=904s&ab_channel=UNIVESP