

Lista de Prioridades

Prof. Ms. Déverson Rogério Rando

Lista de Prioridades

Também denominada *heap*, é composta por um conjunto finito de dados, cada qual com uma chave que determinará sua prioridade. Essa estrutura é um vetor, como uma árvore binária completa: cada nó possui no máximo dois filhos, e os que não têm dois filhos estão no último ou penúltimo níveis. A árvore é sempre preenchida da esquerda para a direita.

Heap

Dado um índice i do vetor, para se descobrir as posições em que se encontram o elemento pai, o filho esquerdo e o filho direito, calcula-se:

$\text{Pai}(i) = \lfloor i / 2 \rfloor$, $\text{Filho Esquerdo}(i) = 2.i$ e $\text{Filho Direito}(i) = 2.i + 1$

Heap Binário

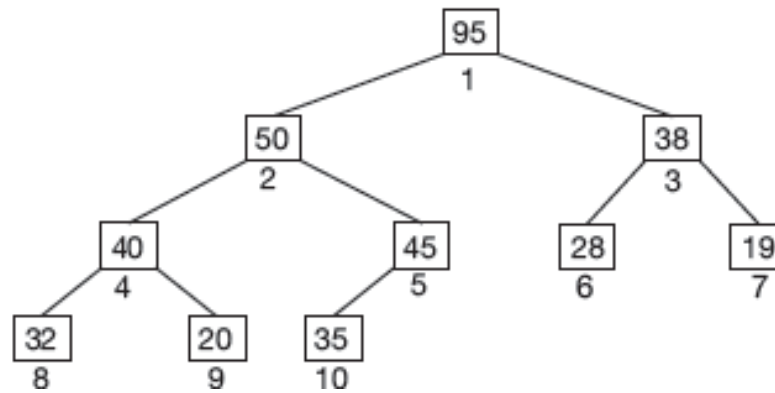
Dois tipos de *heaps* binários: *máximos* e *mínimos*. Todos os nós em um *heap* atendem a determinada propriedade (Cormen, 2002).

- *Heap máximo*: maior elemento sempre está na raiz.

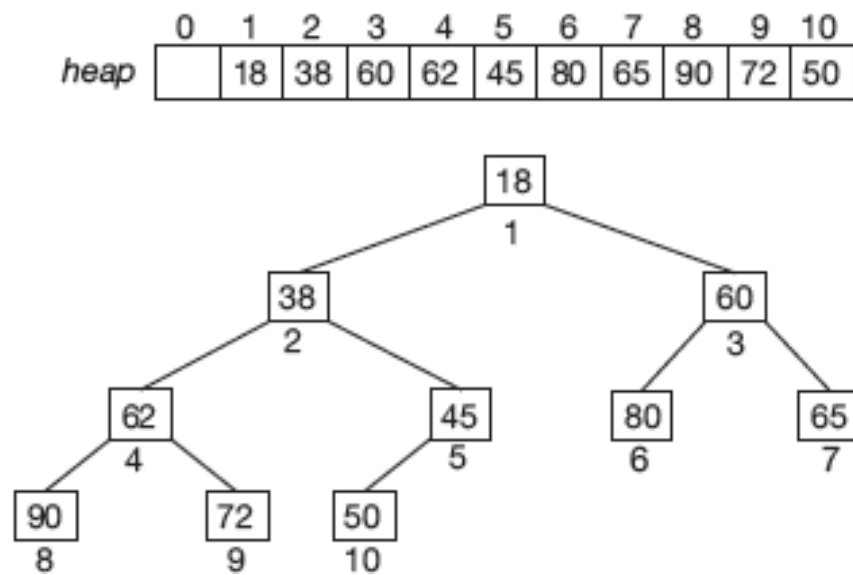
- *Heap mínimo*: menor elemento sempre está na raiz.

Heap máximo

	0	1	2	3	4	5	6	7	8	9	10
heap		95	50	38	40	45	28	19	32	20	35



Heap mínimo



Heap máximo e heap mínimo

- a) o elemento raiz deve ocupar a posição 1 do vetor.
- b) no vetor utilizado para armazenar n elementos a posição inicial é zero (não utilizada); deve ser declarado com $n+1$ posições.
- c) se o *heap* for máximo, a prioridade de um elemento pai deve ser maior que a de seus filhos.
- d) se o *heap* for mínimo, a prioridade de um elemento pai deve ser menor que a de seus filhos.

Operações aplicadas em um *heap* máximo ou mínimo:

- Inserir um elemento na estrutura.
- Remover um elemento da estrutura (o de maior prioridade em um *heap* máximo ou o de menor em um *heap* mínimo).
- Consultar um elemento (de maior prioridade em um *heap* máximo ou de menor em um *heap* mínimo).

Inserção: o elemento deve ser inserido na próxima posição livre, mas antes é verificado se o pai atende à propriedade *heap*. Caso não, é copiado para a posição de inserção e sua posição antiga é avaliada para inserir o novo elemento. A regra é aplicada até se encontrar a posição correta.

Remoção: sempre retira o elemento de maior (*heap* máximo) ou o de menor prioridade (*heap* mínimo), encontrado na raiz. Coloca-se o último elemento no lugar do primeiro.

Após esta movimentação a raiz não atende mais à propriedade *heap*, por isso é aplicado o procedimento de heap novamente.

A operação de *consultar* um elemento acessa o valor da raiz da árvore, que é o primeiro elemento do vetor.

Propriedades a serem obedecidas:

- a) o elemento raiz deve ocupar a posição 1 do vetor;
- b) o vetor utilizado para armazenar n elementos tem a posição inicial zero e, como não será utilizada, deve ser declarado com $n+1$ posições;
- c) em um *heap max-min*, o elemento de maior prioridade ocupa sempre a posição 1 do vetor, e o de menor prioridade ocupa sempre a posição 2 ou 3;



d) em um *heap min-max*, o elemento de menor prioridade ocupa sempre a posição 1, e o de maior prioridade ocupa sempre a posição 2 ou 3 do vetor;

e) cada elemento do vetor, denotado por v_i , corresponde a um nó da árvore binária e encontra-se em um nível. O nível de um nó i é determinado por: $\log_2 i + 1$, para $1 \leq i \leq n$;



f) em um *heap max-min*, os níveis ímpares são chamados de níveis máximos e os pares de mínimos;

g) em um *heap min-max*, os níveis ímpares são chamados de níveis mínimos e os pares de máximos;



h) em um *heap max-min*, se o nível de um nó i é máximo (ímpar), a chave é superior à do seu pai ($v_i > v_{\lfloor i/2 \rfloor}$ quando $i \geq 4$) e inferior à chave do seu avô ($v_i < v_{\lfloor i/4 \rfloor}$ quando $i \geq 4$)

i) em um *heap min-max*, se o nível de um nó i é mínimo (ímpar), a chave é inferior à do pai ($v_i < v_{\lfloor i/2 \rfloor}$ quando $i \geq 4$), e superior à chave do seu avô ($v_i > v_{\lfloor i/4 \rfloor}$ quando $i \geq 4$).

Atividades

1. Implementar o algoritmo para o Heap Máximo.
 1. Inserir
 2. Consultar
 3. Remover
2. Implementar o algoritmo para o Heap Mínimo.
 1. Inserir
 2. Consultar
 3. Remover
3. Entregar até 27/04, enviar por e-mail.