MAC121 - Algoritmos e Estruturas de Dados I

Universidade de São Paulo

Segundo Semestre de 2020

Heapsort

Heapsort

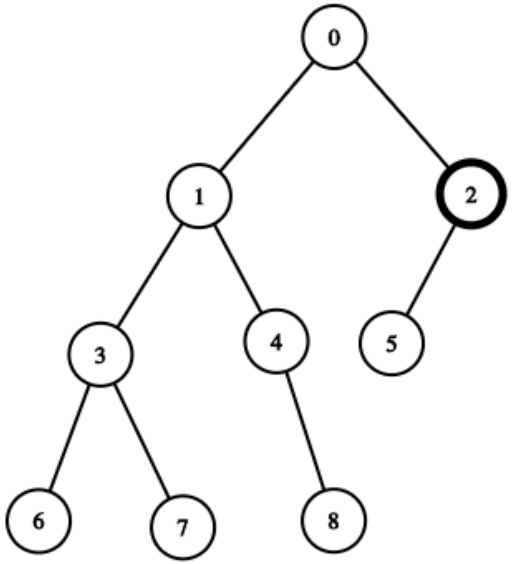
O Heapsort foi inventado em 1964 por J.W.J. Williams. Ainda neste ano, foi melhorado por R.W. Floyd. A ideia do algoritmo se baseia no uso de uma estrutura de dados hierárquica, o heap.

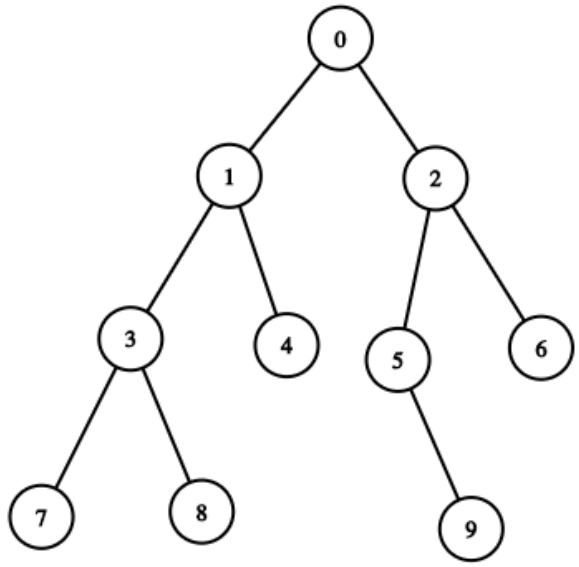
Um heap é uma estutura (binária) hierárquica com as seguintes propriedades:

- É completa até o penúltimo nível;
- Elementos no último nível estão o mais à esquerda possível;
- para todo nó, o valor do nó pai é maior que os dois filhos (max heap).

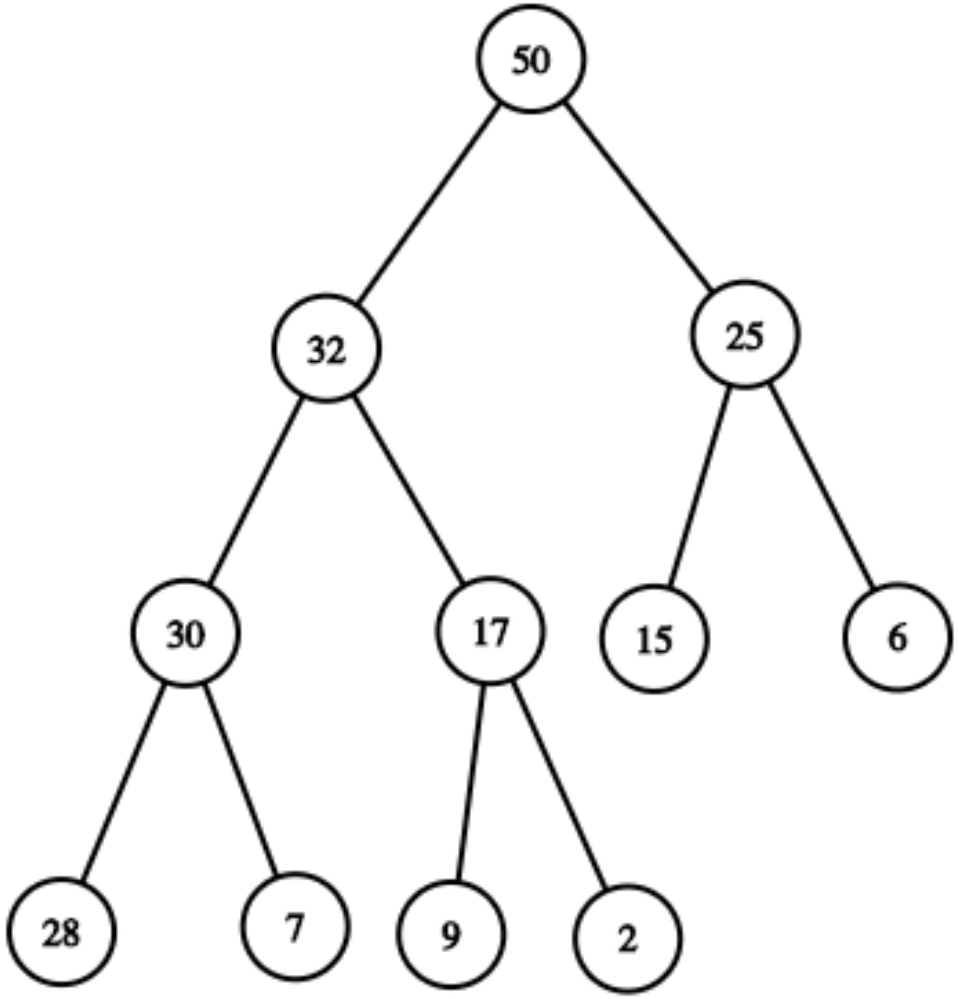
Heap

Não são heaps:





Heap



Em um max heap o maior elemento fica na raiz.

Implementando heaps

Podemos implementar heaps de forma eficiente usando um vetor:

0	1	2	3	4	5	6	7	8	9	10
50	32	25	30	17	15	6	28	7	9	2

Os filhos do elemento que está na posição i estão nas posições $2i + 1$ e $2i + 2$ do vetor.

O pai do elemento que está na posição k está na posição $\frac{k-1}{2}$.

Mas, como usar um heap para ordenar um vetor?

Primeira Ideia: Se eu souber transformar um vetor qualquer em um heap, podemos trocar o primeiro e o último elemento e repetir o processo com um elemento a menos, até termos todo o vetor ordenado.

Heapsort - primeira ideia

```
void heapsort (int *v, int n) {
  int i, aux;
  heapfica (v, n); /* transforma v em heap */
  for (i = n - 1; i > 0; i -) {
    aux = v[0];
    v[0] = v[i];
    v[i] = aux;
    heapfica (v, i);
  }
}
```

Mas, observe que em cada passo apenas o elemento da raiz está no lugar errado e precisará “ser rebaixado” até a posição correta no heap.

Rebaixa

Problema: Faça uma função

```
void rebaixa (int *v, int n, int i);
```

que recebe um vetor v com n elementos, e um índice i de forma que os elementos do vetor no intervalo $[i+1, n-1]$ satisfazem a propriedade do max-heap, rebaixe o elemento até a sua posição correta no heap.

Qual o consumo de tempo da função `rebaixa`?

Heapfica

Mas, e o `heapfica`?

Ideia: basta rebaixar todos os elementos do vetor, a partir do primeiro elemento que tem filhos, indo para trás até chegar na raiz.

```
void heapfica (int *v, int n) {
  int i;
  for (i = (n - 2)/2; i >= 0; i -)
    rebaixa (v, n, i);
}
```

Qual o consumo de tempo?

Para saber mais

- Material sobre Heapsort (P. Feofiloff)
- Livro texto, capítulo 10
- Animação do algoritmo