ORDENAÇÃO EM MEMÓRIA PRIMÁRIA

PARTE 2

1

Classificação por Seleção

- **O Método da Seleção em Árvore - Heapsort**
  - O método utiliza a seleção em árvore para a obtenção dos elementos do vetor na ordem desejada.
  - Ele consiste em duas fases distintas:
    - primeiro é montada uma árvore binária (heap) contendo todos os elementos do vetor, de tal forma que o valor contido em qualquer nó seja maior do que os valores de seus sucessores;
    - na segunda fase, o heap é usado para a seleção dos elementos na ordem desejada.

2

Classificação por Seleção

- **Método Heapsort - Descrição do Algoritmo**
  - Implementação proposta por J. Williams (1964)
  - Dado um vetor de chaves $C_1, C_2, ..., C_N$, consideramos este vetor como sendo a representação de uma árvore binária, usando a seguinte interpretação dos índices das chaves:

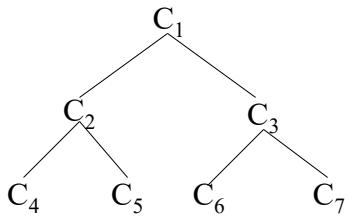
C_i é a raiz da subárvore;
C_{2i} = é a raiz da subárvore da esquerda de C_i
C_{2i+1} = é a raiz da subárvore da direita de C_i

Para i = 1, ..., N div 2

3

Classificação por Seleção

- **Método Heapsort - Descrição do Algoritmo**
  - Exemplificando: dado um vetor $V_{1..7}$, e utilizando a interpretação dada, podemos vê-lo como sendo a representação da seguinte árvore binária:



4

Classificação por Seleção

- **Método Heapsort - Descrição do Algoritmo**
  - O passo seguinte consiste em trocar as chaves de posição dentro do vetor, de tal forma que estas passem a formar uma hierarquia, na qual todas as raízes das subárvores sejam maiores ou iguais a qualquer uma das suas sucessoras, ou seja, cada raiz deve satisfazer as seguintes condições:

$$\begin{cases} C_i \geq C_{2i} \\ C_i \geq C_{2i+1} \end{cases}$$

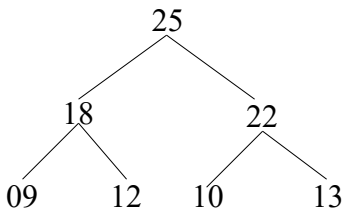
Para i = 1, ..., N div 2.

  - Quando todas as raízes das subárvores satisfazem essas condições, dizemos que a árvore forma um ***heap***.

5

Classificação por Seleção

- **Método Heapsort – Exemplo de um heap**



6

Classificação por Seleção

- **Método Heapsort - Descrição do Algoritmo**
  - O processo de troca de posições das chaves no vetor, de forma que a árvore representada passe a ser um heap, pode ser feito testando-se cada uma das subárvores para verificar se elas, separadamente, satisfazem a condição de heap.
  - Apenas as árvores que possuem pelo menos um sucessor devem ser testada.

7

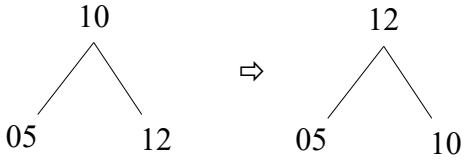
Classificação por Seleção

- **Método Heapsort - Descrição do Algoritmo**
  - A maneira mais simples de realizar o teste é iniciando pela última subárvore, ou seja, aquela cuja raiz está na posição **N div 2** do vetor de chaves, prosseguindo, a partir daí, para as subárvores que a antecedem, até chegar à raiz da árvore.
  - No exemplo, a primeira subárvore a ser testada é aquela cuja raiz é C_3, depois a de raiz C_2 e finalizando com a de raiz C_1.

8

Classificação por Seleção

- **Método Heapsort - Descrição do Algoritmo**
  - Sempre que for encontrada uma subárvore que não forme um heap, seus componentes devem ser rearranjados de modo a formar o heap.



9

Classificação por Seleção

- **Método Heapsort - Descrição do Algoritmo**
  - Pode ocorrer também que, ao rearranjarmos uma subárvore, isto venha a afetar outra, do nível imediatamente inferior, fazendo que esta deixe de formar um heap.
  - Esta possibilidade obriga a verificar, sempre que for rearranjada uma subárvore, se a sucessora do nível abaixo que participou da troca não teve a sua condição de heap desfeita.

10

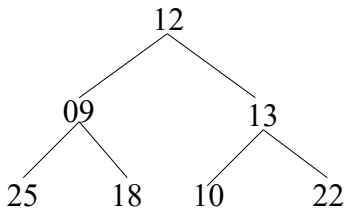
Classificação por Seleção

- **Método Heapsort - Exemplo**

  Seleção da maior chave:

  Vetor inicial: 12 09 13 25 18 10 22

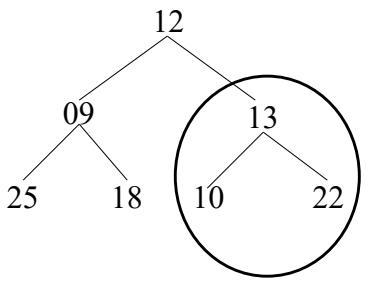
  Representação em árvore:



11

Classificação por Seleção

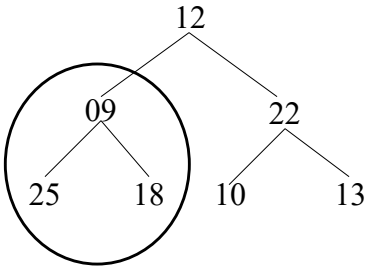
- **Método Heapsort - Exemplo**
  - A transformação dessa árvore em heap inicia pela subárvore cuja raiz é **13**, já que seu índice, no vetor, é **7 div 2 = 3**.



12

Classificação por Seleção

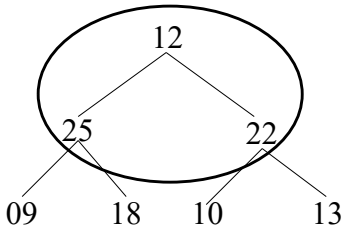
- **Método Heapsort - Exemplo**
  - Vetor e árvore resultantes:12 09 22 25 18 10 13



13

Classificação por Seleção

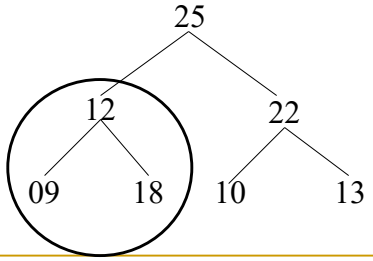
- **Método Heapsort - Exemplo**
  - Vetor e árvore resultantes:12 25 22 09 18 10 13



14

Classificação por Seleção

- **Método Heapsort - Exemplo**
  - Vetor e árvore resultantes:25 12 22 09 18 10 13
  - Neste caso, ocorreu que a transformação da subárvore afetou outra de um nível mais abaixo. A subárvore afetada deve ser rearranjada.

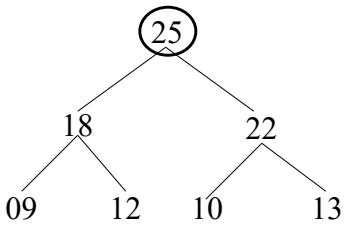


15

Classificação por Seleção

- **Método Heapsort - Exemplo**

  Seleção das chaves: 25 18 22 09 12 10 13
  - Se a chave que está na raiz é a maior chave de todas, então sua posição definitiva correta na ordem crescente é na última posição do vetor, onde ela é colocada, por troca com a chave que ocupa aquela posição.



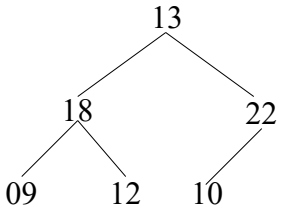
16

Classificação por Seleção

- **Método Heapsort - Exemplo**

  Seleção das chaves: 13 18 22 09 12 10 | 25

  - Com a maior chave já ocupando a sua posição definitiva podemos, a partir de agora, considerar o vetor como tendo um elemento a menos.



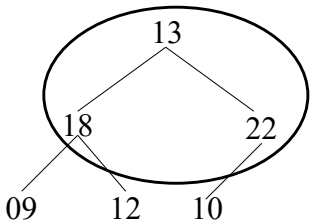
17

Classificação por Seleção

- **Método Heapsort - Exemplo**

  Seleção das chaves: 13 18 22 09 12 10 | 25

  - Para selecionar a próxima chave, deve-se fazer com que a árvore volte a ser um heap.



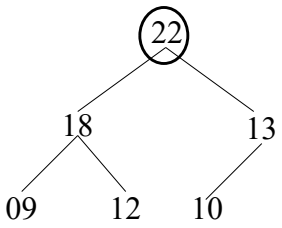
18

Classificação por Seleção

- **Método Heapsort - Exemplo**

  Seleção das chaves: 22 18 13 09 12 10 | 25

  - Novamente a maior chave dentre as restantes aparece na raiz.



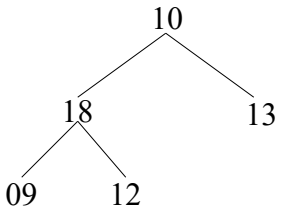
19

Classificação por Seleção

- **Método Heapsort - Exemplo**

  Seleção das chaves: 22 18 13 09 12 10 | 25

  - Vetor e árvore resultantes:10 18 13 09 12 | 22 25
  - A seguir a árvore é novamente rearranjada para formar um heap, o que permitirá selecionar a próxima chave.

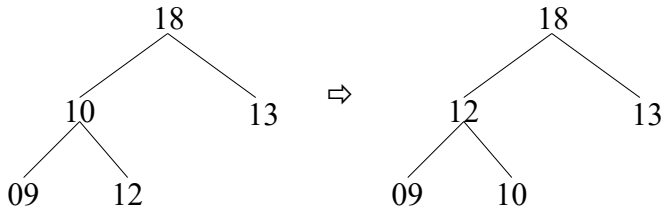


20

Classificação por Seleção

- **Método Heapsort - Exemplo**

  Seleção das chaves: 10 18 13 09 12 | 22 25
  - Vetor intermediário: 18 10 13 09 12 | 22 25
  - Vetor resultante:18 12 13 09 10 | 22 25

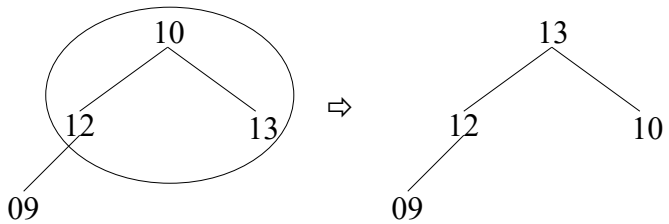


21

Classificação por Seleção

- **Método Heapsort - Exemplo**

  Seleção das chaves: 18 12 13 09 10 | 22 25
  - Vetor intermediário: 10 12 13 09 | 18 22 25
  - Vetor resultante:13 12 10 09 | 18 22 25



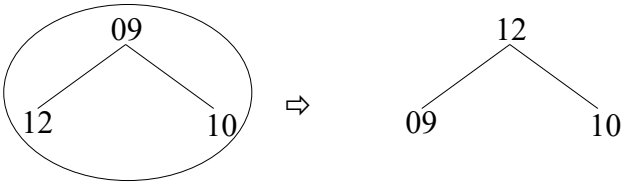
22

Classificação por Seleção

- **Método Heapsort - Exemplo**

  Seleção das chaves: 13 12 10 09 | 18 22 25

  - Vetor intermediário : 09 12 10 | 13 18 22 25
  - Vetor resultante :12 09 10 | 13 18 22 25



23

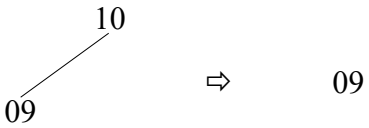
Classificação por Seleção

- **Método Heapsort - Exemplo**

  Seleção das chaves: 10 09 | 12 13 18 22 25

  - Vetor resultante: 09 | 10 12 13 18 22 25

  Vetor final: 09 10 12 13 18 22 25



24

Classificação por Seleção

- **Método Heapsort – Algoritmo em C**

```
public static void heapsort (<tipo_componente>  v [ ], int n) {
   int                          i, r, n1;
  <tipo_componente>   auxkey;

    i = n / 2 - 1;
    for (r = i; r >= 0; r--) {
         heap (v, n, r);
     }
     for (n1 = n-2; n1 >= 0; n1--) {
         auxkey = v [0];
         v[0] = v[n1+1];
         v[n1+1] = auxkey;
         heap (v, n1, 0);
     }
}
```

25

Classificação por Seleção

- **Método Heapsort – Algoritmo em C (cont.)**

```
void heap (<tipo_componente> v [ ] , int n, int r) {
    int                          i, h, troca;
   <tipo_componente>   auxkey;

    i = r; troca = 1;
    while (troca == 1) {
         troca = 0;
         if (keyval(v, n, 2*i+1) > keyval(v, n, 2*i+2))
              h = 2*i+1;
         else
              h = 2*i+2;
         if (v[i] < keyval(v, n, h)) {
               auxkey = v[i]; v[i] = v[h]; v[h] = auxkey; i = h; troca = 1;
         }
     }
}
```

26

Classificação por Seleção

- **Método Heapsort – Algoritmo em C (cont.)**

```
<tipo_componente> keyval (<tipo_componente> v [ ], int n, int i) {
    if (i > n)
        return  minkey;
    else
        return v[i];
}
```

OBS: ***minkey*** corresponde ao menor valor de chave possível de ser representado. Por exemplo, em Java, se as chaves forem do tipo int, então ***minkey*** seria igual a Integer.MIN_VALUE.

27

Classificação por Seleção

- **Sugestão de leitura:**
  - Capítulo 6 do livro "CORMEN, Thomas H. **Algoritmos:** teoria e prática. 3. ed. Rio de Janeiro: Elsevier, 2012. "

28