Algoritmia e

Programação

Operações em vectores.



Ordenar os valores num vector

•

•

•

•

Em certas aplicações convém ter os valores ordenados (por ordem crescente ou decrescente) num vector.

No entanto, os valores podem ter sido inseridos fora de ordem:

Nesses casos é necessário proceder à **ordenação** do vector.

Há vários algoritmos de ordenação, alguns mais eficientes do que outros. Vamos analisar a **ordenação por selecção**.



**APROG - LETI**

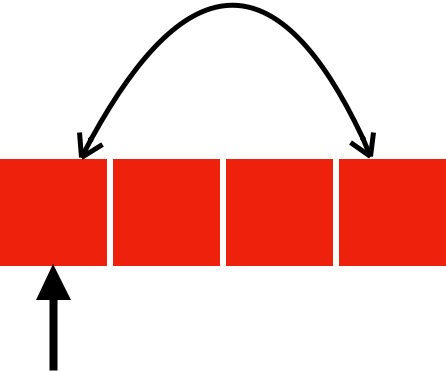
Ordenação por selecção: estratégia

•

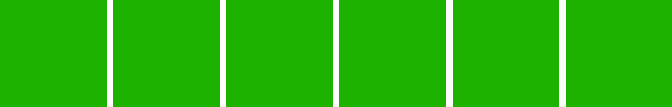
•

Seleccionar o menor dos elementos não-ordenados, e trocar com o elemento na primeira posição do vector.

Prosseguir da mesma forma para as posições seguintes, criando uma sequência ordenada "à esquerda" a partir dos elementos desordenados "à direita".



**Valores já ordenados**



**Troca a ser realizada**



2 4 7 9 10 11 45 33 27 19 23 38

**Posição a ser ordenada**

* Quando se processar a penúltima posição do vector, o vector estará ordenado.



**APROG - LETI**

Ordenação por selecção: análise

* É necessário tratar cada posição do vector antes de se passar para a posição seguinte.
  + Percorrer desde a primeira até à penúltima posições do vector.
* Para cada posição do vector a tratar é necessário procurar o menor valor dos que ainda não estão ordenados.
  + Obriga a percorrer todas as posições que ainda não estão ordenadas, pois a posição do menor é desconhecida.



**APROG - LETI**

Ordenação por selecção: algoritmo



Percorrer do primeiro ao penúltimo elemento Percorrer do elemento seguinte até ao último

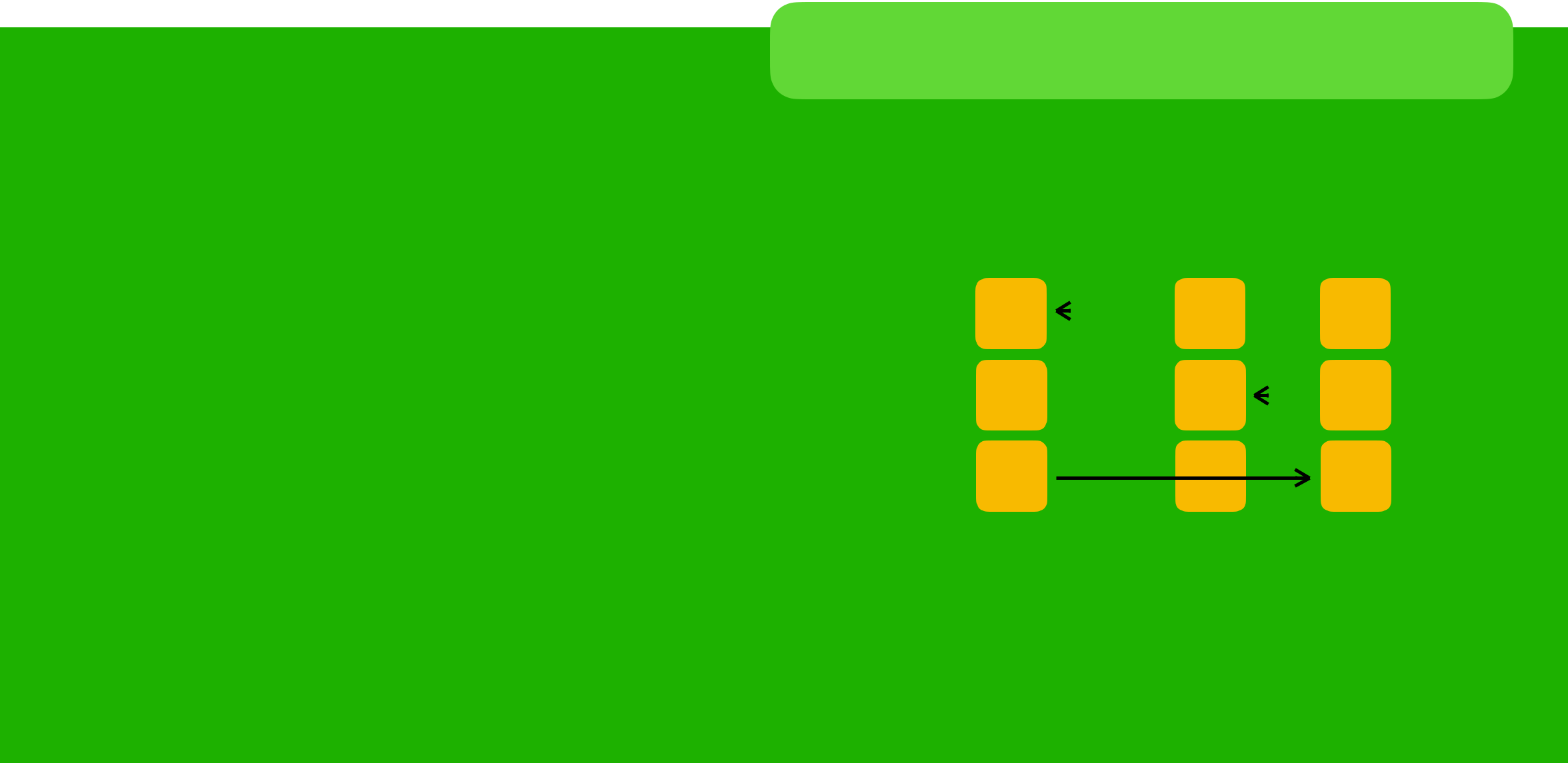
Se o elemento à direita < elemento a ordenar então Troca o elemento à direita com o elemento a ordenar



**APROG - LETI**

O vector v tem n elementos: v[0] … v[n-1]

Ordenação por selecção: algoritmo



PARA i = 0 ATÉ n-2 FAZER

PARA j = i+1 ATÉ n-1 FAZER SE v[i] > v[j] ENTÃO

aux = v[i]

v[i] = v[j]

v[j] = aux

FIM SE

FIM PARA

FIM PARA

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **aux** |  | **v [ i ]** |  | **v [ j ]** |  |
| 9 |  | 9 | 5 | |  |
|  |  |
| 9 | 5 | |  | 5 |  |
|  |  |
| 9 | 5 | | 9 | |  |



**APROG - LETI**

Pesquisar num vector

* Procurar um valor num vector desordenado obriga a percorrer sequencialmente o vector:

•

•

a pesquisa pára quando o valor é encontrado, ou

a pesquisa pára quando se atinge o fim do vector sem se encontrar o valor.

* No entanto, a pesquisa pode ser optimizada se o vector estiver ordenado.
  + Vamos estudar a **pesquisa binária**.



**APROG - LETI**

Pesquisar binária: estratégia

* O primeiro elemento a ser pesquisado é o que se encontra na **posição central** do vector.
  + Se não for o valor a procurar, temos o vector dividido em duas metades:
    - a dos valores menores do que o central, e
    - a dos valores maiores do que o central.
  + A pesquisa prossegue na metade em que o valor a procurar poderá se encontrar.
    - Pesquisa-se na posição central dessa metade, e prossegue-se assim, com secções cada vez menores.



**APROG - LETI**

Pesquisa binária: algoritmo



O vector v tem n elementos: v[0] … v[n-1].

Procuramos o valor T.

SENÃO SE(v[i] > T) ENTÃO

l = 0 r = m - 1

r = n-1 SENÃO

encontrou = 0 ESCREVER(m)

encontrou = 1

ENQUANTO(l < r E encontrou = 0) FIM SE

FAZER FIM ENQUANTO

i = (l + r) / 2

SE(v[i] < T) ENTÃO SE(encontrou = 0) ENTÃO

l = m + 1 ESCREVER("T não encontrado")



**APROG - LETI**