

Ficha Prática 2

António Nestor Ribeiro, José Creissac Campos
{anr,jose.campos}@di.uminho.pt
PIIV (LESI)

2002/03

Objectivos

1. Apresentação da API Java
2. Packages `java.lang` e `java.util`.
3. Arrays: algoritmos de ordenação e pesquisa.

Exercícios

1. Implemente a ordenação crescente de um array de inteiros usando um algoritmo de trocas directas. Este algoritmo baseia-se na sucessiva identificação de mínimos, sendo que determina para cada posição o mínimo valor que a ela corresponde.

O primeiro mínimo fica na posição 0, o mínimo do array de 1 a `length-1` fica na posição 1 e assim sucessivamente.

A solução passa por efectuar várias passagens no array para encontrar mínimos. O número de passagens é igual a `length-1` e no final da passagem K obtemos o mínimo a colocar na posição K-1.

Não é necessário fazer a passagem final, uma vez que para obter o mínimo para `length-2` as duas últimas posições ficam ordenadas.

Exemplo de uma iteração:

Encontrar o 0

```

    0 | | | | | | | | | 10
-----o---o---o---o---o---o---o---o---o---o---o---o---
|-5 |-1 | 0 | 2 | 4 | 9 |10 |13 |15 |20 |27 |
-----
    ^           ^           ^
    |           |           |
li = 0         meio = 5     ls = 10
```

Depois da primeira partição

```

    0 | | | | 5 | | | | |
-----o---o---o---o---o---o---o---o---o---o---o---o---
|-5 |-1 | 0 | 2 | 4 | 9 |10 |13 |15 |20 |27 |
-----
    ^     ^     ^
    |     |     |
li = 1 meio =2     ls = 5
```

3. Considere a declaração:

```
int[] [] turmas = {{.....},
                  {.....},
                  {..}
                  };
```

correspondente a um *array* bidimensional em que cada linha contém os números dos alunos de um dado turno prático de PPIV. Escreva código Java para:

- Calcular o número total de alunos inscritos.
- Calcular qual a turma mais pequena.
- Calcular o número do aluno mais antigo.
- Ordenar as turmas por ordem crescente de tamanho.