

Unidade IV:

Algoritmo de Inserção

Exercício (1):

Uma dúvida básica sobre o operador AND pode prejudicar a compreensão do nosso algoritmo. Assim, o que será escrito na tela pelo programa abaixo?

```
class ExercicioDuvidaAND {  
    public static boolean m1(){  
        System.out.println("m1");  
        return false;  
    }  
    public static boolean m2(){  
        System.out.println("m2");  
        return true;  
    }  
    public static void main (String[] args) {  
        System.out.println("inicio");  
        boolean and = m1() && m2();  
        System.out.println("fim: " + and);  
    }  
}
```

Resposta:

Inicio
M1
fim: false

Exercício (2):

Mostre todas as comparações entre elementos do array para os arrays abaixo:

1	2	3	4	5	6
---	---	---	---	---	---

6	5	4	3	2	1
---	---	---	---	---	---

Resposta:

1.	1 2 3 4 5 6
	1 > 2 = false
	2 > 3 = false
	3 > 4 = false
	4 > 5 = false
	5 > 6 = false
2.	6 5 4 3 2 1
	6 > 5 = true, então: 5 6 4 3 2 1
	6 > 4 = true, então: 4 5 6 3 2 1
	6 > 3 = true, então: 3 4 5 6 2 1
	6 > 2 = true, então: 2 3 4 5 6 1
	6 > 1 = true, então: 1 2 3 4 5 6

Exercício (3):

Mostre todas as comparações e movimentações do algoritmo anterior para o array abaixo:

12	4	8	2	14	17	6	18	10	16	15	5	13	9	1	11	7	3
----	---	---	---	----	----	---	----	----	----	----	---	----	---	---	----	---	---

Resposta:

12 4 8 2 14 17 6 18 10 16 15 5 13 9 1 11 7 3
12 > 4 = true, então: 4 12 8 2 14 ...
12 > 8 = true, então: 4 8 12 2 14 ...
12 > 2 = true, então: 2 4 8 12 14 ...
12 > 14 = false
14 > 17 = false
17 > 6 = true, então: 2 4 6 8 12 14 17 18 ...
17 > 18 = false
18 > 10 = true, então: 2 4 6 8 10 12 14 17 18 16 15 ...
18 > 16 = true, então: 2 4 6 8 10 12 14 16 17 18 15 ...
18 > 15 = true, então: 2 4 6 8 10 12 14 15 16 17 18 5 13 9 ...
18 > 5 = true, então: 2 4 5 6 8 10 12 14 15 16 17 18 13 9 ...
18 > 13 = true, então: 2 4 5 6 8 10 12 13 14 15 16 17 18 9 ...
18 > 9 = true, então: 2 4 5 6 8 9 10 12 13 14 15 16 17 18 ...
18 > 1 = true, então: 1 2 4 5 6 8 9 10 12 13 14 15 16 17 18 11 7 3
18 > 11 = true, então: 1 2 4 5 6 8 9 10 11 12 13 14 15 16 17 18 7 3
18 > 7 = true, então: 1 2 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 3
18 > 3 = true, então: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

Exercício (4):

Uma forma de melhorar o Algoritmo de Inserção é considerar a pesquisa binária para procurar a posição em que o novo elemento será inserido na lista ordenada. Nesse caso, realizamos $O(\lg m)$ comparações, onde m é o tamanho da lista ordenada, para encontrar a posição de inserção. Em seguida, em uma estrutura de repetição, movemos em uma unidade todos os elementos já ordenados e cuja posição é maior ou igual a de inserção. Implemente o Algoritmo de Inserção com Pesquisa Binária.

Resposta:

O algoritmo se apresenta na pasta **exercícios_praticos**.

Exercício (5):

Quando os elementos estão ordenados de forma decrescente tanto o Seleção como o Inserção realizam (imagem) comparações. Nesse caso, qual dos dois algoritmos executará mais rápido? Justifique sua resposta.

$$C(n) = \frac{(n-1)(n)}{2} = \frac{n^2}{2} - \frac{n}{2}$$

Resposta:

Eles iram realizar o mesmo número de comparações então serão igualmente rápidos.