

Pesquisa Binária e Variações

1 – Construa um método `criaArrayCom(int valor, int dimensao, boolean diferentes)` que recebe um valor inteiro e uma dimensão. Deve devolver um array de inteiros ordenado (todos diferentes se o parâmetro `diferentes` for verdadeiro), com a dimensão especificada, que inclui o valor indicado.

2 – Construa um método `criaArraySem(int valor, int dimensao, boolean diferentes)`, que recebe um valor inteiro e uma dimensão. Deve devolver um array de inteiros ordenado (todos diferentes se o parâmetro `diferentes` for verdadeiro), com a dimensão especificada, que não inclui o valor indicado.

Use os métodos criados nas alíneas anteriores para testar a resolução das alíneas seguintes.

3 – Construa um método que efectua uma pesquisa binária de forma recursiva, devolvendo `true` ou `false` conforme o valor seja encontrado ou não.

4 – Construa um método que efectua uma pesquisa binária de forma iterativa, devolvendo `true` ou `false` conforme o valor seja encontrado ou não.

5 – Construa um método que efectua uma pesquisa binária. Este método deve devolver a posição em que o valor procurado se encontra, ou então -1 caso este não esteja no array indicado. Teste o método, conforme o indicado nos exercícios anteriores.

6 – Construa um método que efectua uma pesquisa binária. Este método deve devolver a posição em que o valor procurado se encontra, ou então um valor negativo (-X) caso este não esteja no array indicado. O valor de `abs(X+1)` deve indicar uma posição em que o valor procurado poderia ser inserido para preservar a ordem. Teste o método, conforme o indicado nos exercícios anteriores.

Exemplo:

Array: {3,7,12,15}

Valor procurado=15 resultado= 3

Valor procurado=3 resultado= 0

Valor procurado=1 resultado= -1 (o valor deveria ser inserido na posição 0)

Valor procurado=4 resultado= -2 (o valor deveria ser inserido na posição 1)

Valor procurado=10 resultado= -3 (o valor deveria ser inserido na posição 2)

Valor procurado=13 resultado= -4 (o valor deveria ser inserido na posição 3)

Valor procurado=16 resultado= -5 (o valor deveria ser inserido na posição 4)

Para resolver os exercícios seguintes, tenha em consideração o método de pesquisa binária e as variantes apresentadas nas alíneas anteriores. Todos os exercícios devem ser resolvidos com algoritmos com complexidade $O(\log N)$.

7 – Construa um método que recebe um array ordenado de inteiros, todos diferentes, e um valor, e devolve a percentagem de valores do array que são menores do que o valor indicado.

Exemplo:

Array: {3,7,12,15}

Valor =15 resultado= 0.75

Valor =14 resultado= 0.75

Valor =3 resultado= 0.0

Valor =1 resultado= 0.0

Valor =100 resultado= 1.0

8 – Construa um método que recebe um array ordenado de inteiros, todos diferentes, e dois valores que definem uma intervalo. O método deve indicar quantos valores do array se encontram dentro do intervalo especificado.

Exemplo:

Array: {3,7,12,15}

Valores =(0,15) resultado= 4

Valores=(3,7) resultado= 2

Valores =(4,14) resultado= 2

Valores =(4,5) resultado= 0

Valores =(0,100) resultado= 4

9 – Construa um método que recebe um array ordenado de inteiros, e um valor. O método deve indicar se esse valor se encontra repetido no array.

Exemplo:

Array: {3,3,7,12,12,15}

Valor =15 resultado= false

Valor =14 resultado= false

Valor =12 resultado= true

Valor =3 resultado= true

10 - Construa um método que devolve recebe por parâmetro um array ordenado de inteiros, não repetidos, bem como um inteiro Z, e devolve o maior elemento do array menor do que Z (ou Z se esse elemento não existir).

Exemplo:

Array: {3,7,12,15}

Valor =15 resultado= 12

Valor =14 resultado= 12

Valor =3 resultado= 3

Valor =1 resultado= 1

Valor =100 resultado= 15

11 – Considere um método que recebe um array de inteiros no qual os números estão dispostos da seguinte forma: todos os números negativos se encontram em posições maiores do que os números positivos, e todos os número positivos e negativos se encontram ordenados entre si.

Exemplo: {3,6,8, -10,-3,-2,-1}

Construa um método que procura um número no array.

12 – Construa um método que recebe um array ordenado de inteiros, não repetidos, e que devolve o índice da primeira posição na qual o valor guardado é superior ao índice (ou -1 se esta posição não existir).

Exemplos:

Array: {3,7,12,15} Resultado=0

Array: {-3,1,7,12,15} Resultado=2

Array: {-15,-14,1,2,3,4} Resultado= -1