Estrutura de Dados

Recursividade

O algoritmo recursivo abaixo retorna o maior elemento do vetor.

Realiza comparação entre os conteúdos

```
public static int maior(int v[], int inicio, int fim) {
int meio = (inicio + fim) / 2;
int n1, n2;
if (meio > inicio) {
n1 = maior(v, inicio, meio);
n2 = maior(v, meio + 1, fim);
} else {
n1 = v[inicio];
n2 = v[fim];
}
if (n1 > n2)
return n1;
else
return n2;
}
```

Exemplo:

Inicia os tamanho do vetor, o inicio e o fim do vetor
Define a metade do vetor
Cria as variáveis n1,n2
Se o meio for maior que o inicio faz:
    Por meio de divisão e conquista é feito a atribuição para n1 e n2 a qual será feita a recursão para encontrar esse valor.
Caso contrario
    N1 recebe o elemento do v[inicio]
    N12recebe o elemento do v[fim]
Se n1 for maior que n2
    Retorna n1
senão
    Retorna n2

*Código-fonte fornecido pela Mestre-Doutoranda **Eliane Oliveira Santiago.**

Teste de MESA

Em um vetor (v[3]) de 3 posições, onde o inicio é 0 e fim é 2. `maior(v[3],0, 2);`

Atribui respectivamente os valores 6, 4, 7

V[3]	0	1	2
Valor	6	4	7

*maior(v[3],0,2)	Meio	Inicio	Fim	N1	N2
Valor	(0+2)/2 = 1	0	2	-	-

Se meio > inicio // 1 > 0

Caso1 | Caso2

*maior(v[3],0,2)	Meio	Inicio	Fim	N1	N2
Valor	1	0	2	maior(v[3],0, 1);	maior(v[3], 1+1 , 2);

Nesse momento começa agir a recursividade, onde o método **maior** chama o próprio método para encontrar o maior valor por meio de divisão e conquista.

Caso1

maior(v[3],0,1)	Meio	Inicio	Fim	N1	N2
Valor	(0+1)/2 = 0	0	1	-	-

Se meio > inicio // 0> 0

Senão

Variável	N1	N2
Valor	n1 = v[inicio];	n2 = v[fim];

Variável	N1	N2
Valor	n1 = 6;	n2 = 4;

N1 recebe o elemento contido na posição do vetor v[0]

N2 recebe o elemento contido na posição do vetor v[1]

Respectivamente 6 e 4.

Se n1 > n2 // 6> 4

Variável	N1
Retorna	n1 = 6;

Até o momento 6 é o maior elemento obtido na recursividade maior(v[3],0, 1);

Como foi feita o a recursiva para n1 e para n2(divisão e conquista) o mesmo procedimento se aplicará para n2.

```
n1 = maior(v, inicio, meio);
n2 = maior(v, meio + 1, fim);
```

*A divisão de ½ retorna 0 pois a variável ‘meio’ é int.

Caso2

maior(v[3],0,1)	Meio	Inicio	Fim	N1	N2
Valor	(2+2)/2 = 2	2	2	-	-

Se meio > inicio // 2> 2

N1 continua valendo 6(do lado de fora dessa recursividade)

Senão

Variável	N1	N2
Valor	n1 = v[inicio];	n2 = v[fim];

Variável	N1	N2
Valor	n1 = 6;	n2 = 4;

N1 recebe o elemento contido na posição do vetor v[2]

N2 recebe o elemento contido na posição do vetor v[2]

Respectivamente 7 e 7.

Se n1 > n2 // 7> 7

Senão

Variável	N2
Retorna	n2 = 7;

Até o momento 7 é o maior elemento obtido na recursividade maior(v[3], 1+1 ,2);

Agora que foi obtido os dois maiores valores sendo eles n1 = 6 e n2 =7 é feita a comparação entre ambos

Se n1 > n2 // 6> 7 ...

Senão

Variável	N2	v[2]
Retorna	n2 = 7;	7

*Retorna o elemento contido no vetor. Procedimento será feito até que encontre o maior elemento.

Material para Hands-On

Código

Principal_maior_Recursivo.java

```
public class Principal_maior_Recursivo {

    public static void main(String[] args) {


        int vetor[] = new int [3];
        vetor[0]= 6;
        vetor[1]= 4;
        vetor[2]= 7;

        System.out.println(MaoirRecursivo.maior(vetor, 0, vetor.length-1));
    }
}
```

Código

MaoirRecursivo.java

```java
public class MaoirRecursivo {

    public static int maior(int v[], int inicio, int fim) {
        int meio = (inicio + fim) / 2;
        int n1, n2;
        if (meio > inicio) {
            n1 = maior(v, inicio, meio);
            n2 = maior(v, meio + 1, fim);
        } else {
            n1 = v[inicio];
            n2 = v[fim];
        }
        if (n1 > n2)
            return n1;
        else
            return n2;
    }
}
```

Referencias.

- Código-fonte fornecido pela Mestre-Doutoranda **Eliane Oliveira Santiago, Exercicio 05 de ‘Laboratório de Estruturas de Dados Lineares e Recursividade’.**
- Ferramenta para Debug – Eclipse IDE for Java Developers - 2020-06.