

Marcos Geraldo Braga Emiliano

19.1.4012

Exemplo de teste para demonstrar o algoritmo

adapada

O algoritmo que eu imaginei se consiste em, avaliar a partir da esquerda um elemento i , que é comparado com um elemento j que se inicia a partir da direita, e para encontrar os palíndromos realizar a comparação dos elementos i e j e caso eles sejam iguais o valor de i é acrescido e o de j decrescido, para caminhar na palavra de entrada, e é retornado o maior palíndromo encontrado.

Versão Recursiva

```
func palindro(palavra[1..n], i, j){
    if(j<i)
        return 0;
    if( i == j){
        if(palavra[i] == palavra[j])
            return 1;
        else
            return 0;
    }
    Int max=0;
    If(palavra[i] == palavra[j])
        Aux1 = palindro(palavra[1..n], i-1, j-1) + 2;
    else
        Aux2 = palindro(palavra[1..n], i-1, j);
        Aux3 = palindro(palavra[1..n], i, j-1);
    max = max(Aux1,Aux2,Aux3);
    return max;
}
```

Versão de Programação Dinâmica

Matriz de resultado >>>>>>>>>> Result[0..n, 0..n]

Result[0,0]=0#Caso Base

```
for ( i=0 , i<=n , i++){
```

```
for ( j=0 , j<=n , j++){
```

```
if( palavra[i] == palavra[n-j] ) #Compara os caracteres
```

```
if( i == j )
```

```
Result[i,j] = Result[i-1,j-1]+1 #Palindromo impar
```

else

```
Result[i,j] = Result[i-1,j-1]+2 # Palindromo par
```

}

}