

Lista 12**Busca binária e Custo Computacional****Questão 1**

Considere o problema de determinar o valor de um elemento máximo de um vetor $v[0..n-1]$. Considere a função abaixo.

```
1 int maximo(int n, int *v){  
2     int i, x;  
3     x = v[0];  
4     for (i = 1; i < n; i++)  
5         if (x < v[i])  
6             x = v[i];  
7     return x;  
8 }
```

- (a) A função maximo acima resolve o problema?
- (b) Faz sentido trocar $x = v[0]$ por $x = 0$?
- (c) Faz sentido trocar $x = v[0]$ por $x = \text{INT_MIN}$?
- (d) Faz sentido trocar $x < v[i]$ por $x \leq v[i]$?

Questão 2

Considere o código abaixo:

```
1 int buscaR2(int n, int *v, int x){  
2     if (v[n-1] == x)  
3         return 1;  
4     else  
5         return buscaR2(n-1, v, x);  
6 }
```

- (a) Qual é o comportamento da função `buscaR2()`?
- (b) Quantas vezes a função `buscaR2()` é chamada?
- (c) O que acontece quando x não está no vetor v ?
- (d) Qual o custo computacional dessa função?

Questão 3

Considere a função de busca binária abaixo:

```
1 int busca_binaria(int n, int *v, int x){
2     int esq, dir, meio;
3     esq = -1;
4     dir = n;
5     while (esq < dir - 1) {
6         meio = (esq + dir) / 2;
7         if (v[meio] < x)
8             esq = meio;
9         else
10            dir = meio;
11    }
12    return dir;
13 }
```

Considere os valores $v[i] = i$, para $i = 0, 1, \dots, n$.

- (a) Execute a função com $n = 9$ e $x = 3$;
- (b) Execute a função com $n = 14$ e $x = 7$;
- (c) Execute a função com $n = 15$ e $x = 7$.

Questão 4

Execute uma função de busca binária com $n = 16$. Quais os possíveis valores de $m = n/2$ na primeira iteração? Quais os possíveis valores de $m = n/2$ na segunda iteração? Na terceira? Na quarta?

Questão 5

Suponha que cada elemento do vetor $v[0..n-1]$ é uma cadeia de caracteres (ou seja, temos uma matriz de caracteres). Suponha também que o vetor está em ordem lexicográfica.

- (a) Escreva uma função eficiente, baseada na busca binária, que receba uma cadeia de caracteres x e devolva um índice k tal que x é igual a $v[k]$. Se tal k não existe, a função deve devolver -1 .
- (b) Qual é a complexidade de tempo da função escrita no item (a)?

Questão 6

Qual o menor valor de n tal que um programa com tempo de execução $100n^2$ é mais rápido que um programa cujo tempo de execução é $2n$, supondo que os programas foram implementados no mesmo computador?

Questão 7

Expresse a função $n^3/1000 - 100n^2 - 100n + 3$ na notação O .

Questão 8

Suponha que estamos comparando as implementações de dois métodos para resolver um problema em um mesmo computador. Para entradas de tamanho n , o primeiro método gasta $8n^2$ passos enquanto que o segundo método gasta $64n \log n$ passos. Para quais valores de n primeiro método é melhor que o segundo?

Questão 9

Suponha que você tenha algoritmos com os seis tempos de execução listados abaixo. Suponha que você tenha um computador capaz de executar 10^{10} operações por segundo e você precisa computar um resultado em no máximo uma hora de computação. Para cada um dos algoritmos, qual é o maior tamanho da entrada n para o qual você poderia receber um resultado em uma hora?

	maior valor de n ?
n^2	
n^3	
$100n^2$	
$n \log_2 n$	
2^n	
2^{2^n}	

Questão 10

Faça uma função `merge()` que recebe dois vetores **ordenados** **A**, de tamanho **n**, e **B**, de tamanho **m**, e retorna os valores de **A** e **B** em ordem em um terceiro vetor **C**. Sua função deverá possuir a seguinte assinatura:

```
1 int* merge(int* A, int* B, int n, int m);
```

Faça uma análise teórica de pior e melhor caso do seu algoritmo.