

FCT/Unesp – Presidente Prudente
Projeto e Análise de Algoritmos
Prof. Danilo Medeiros Eler

Exercícios Aula 02

<https://danioloeler.github.io/teaching/PAA2020/index.html>

1) Para cada um dos trechos de código abaixo, analise o tempo estimado de execução no **melhor** e no **pior** caso, considerando o modelo RAM. Considere que as variáveis **n**, **m** e **vetor** sejam dados de entrada.

a)

```
int soma = 0;
for (int i=0; i<n; i++)
    soma = soma + i;
```

b)

```
int soma1 = 0;
int soma2 = 0;
for (int i=0; i<n; i++){
    soma1 = soma1 + 1;
    soma2 = soma2 + i;
}
```

c)

```
int soma = 0;
for (int i=0; i<n; i++){
    if ( vetor[i] % 2 == 0) //se for par
        soma = soma + vetor[i];
}
```

d)

```
int soma1 = 0;
for (int i=0; i<n; i++){
    soma1 = soma1 + 1;
}
for (int j=0; j<n;j++){
    soma1 = soma1 + j;
}
```

e)

```
int soma = 0;
for (int i=0; i<n; i++){
    for (int j=0; j<n; j++){
        soma = soma + 1;
    }
}
```

f)

```
int soma = 0;
for (int i=0; i<n; i++){
    for (int j=0; j<m; j++){
        soma = soma + 1;
    }
}
```

g)

```
int menor = MAIOR-INTEIRO;
for (int i=0; i<n; i++){
    if (vetor[i] < menor)
        menor = vetor[i];
}
```

h)

```
int v[][] = new int[n][n];

for (int i=0; i<n; i++){
    for (int j=0; j<n; j++){
        v[i][j] = i * j;
    }
}
```

i)

```
int menor = MAIOR-INTEIRO;
for (int i=0; i<n; i++){
    if (vetor[i] < menor)
        menor = vetor[i];
}
if (menor < 0){
    for (int i=0; i<n; i++){
        menor = menor * (i+1);
    }
}
```

j)

```
int menor = MAIOR-INTEIRO;
for (int i=0; i<n; i++){
    if (vetor[i] < menor)
        menor = vetor[i];
}
if (menor < 0){
    for (int i=0; i<n; i++){
        menor = menor * (i+1);
    }
}
else if (menor > 0){
    for (int i=0; i<n*n; i++)
        printf(“%d\n”, menor);
}
else {
    printf(“%d\n”, menor);
}
```

2) Dado o método de busca a seguir, analise o tempo estimado de execução no **melhor** e no **pior** caso para cada um dos trechos de código, considerando o modelo RAM. Lembre que **size()** é um método que retorna a quantidade de elementos de uma lista.

```
Pessoa busca(String nome){
    for (int i = 0; i< pessoas.size(); i++){
        if (pessoas.get(i).getNome().equals(nome))
            return pessoas.get(i);
    }
    return null;
}
```

a)

```
void exibir(String nome){
    Pessoa p = busca(nome);
    if (p != null){
        p.exibirDados();
    }
    else{
        System.out.println("Pessoa não encontrada");
    }
}
```

b)

```
void exibir(String nome){
    if (busca(nome) != null){
        busca(nome).exibirDados();
    }
    else{
        System.out.println("Pessoa não encontrada");
    }
}
```

c)

```
void atualizar(String nome, int idade, float salario){
    Pessoa p = busca(nome);
    if (p != null){
        p.setIdade(idade);
        p.setSalario(salario);
    }
    else{
        System.out.println("Pessoa não encontrada");
    }
}
```

d)

```
void atualizar(String nome, int idade, float salario){
    if (busca(nome) != null){
        busca(nome).setIdade(idade);
        busca(nome).setSalario(salario);
    }
    else{
        System.out.println("Pessoa não encontrada");
    }
}
```