

Instituto Federal de Ciência e Tecnologia da Bahia

Projeto e Análise de Algoritmos

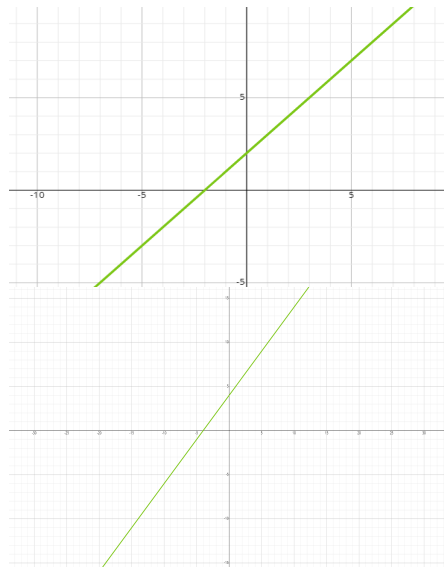
Prof. Felipe Oliveira dos Santos

Exercícios

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.

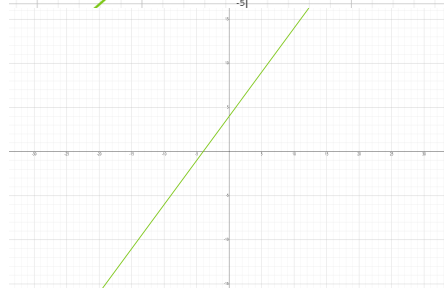
a)

```
int soma = 0; +1
for (int i=0; i<n; i++) n +1
    soma = soma + i;
TOTAL: n+2
```



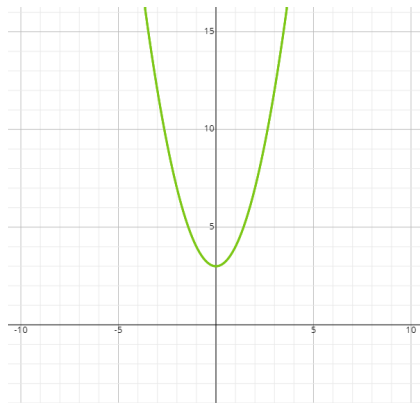
b)

```
int soma1 = 0; +1
int soma2 = 0; +1
for (int i=0; i<n; i++){ n+2
    soma1 = soma1 + 1;
    soma2 = soma2 + i;
}
TOTAL: n+4
```



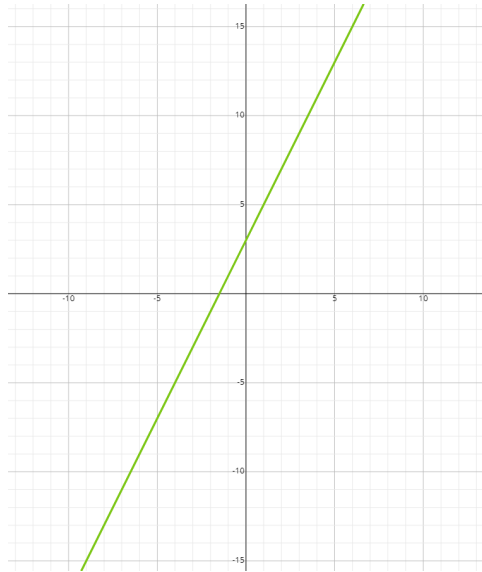
c)

```
int soma = 0; +1
for (int i=0; i<n; i++){ n+2
    if ( vetor[i] % 2 == 0) //se for par
        soma = soma + vetor[i];
}
TOTAL: n^2+3
```



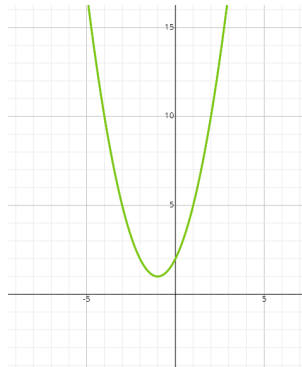
d)

```
int soma1 = 0; +1
for (int i=0; i<n; i++){ n+1
    soma1 = soma1 + 1;
}
for (int j=0; j<n; j++){ n+1
    soma1 = soma1 + j;
}
TOTAL: 2n+3
```



e)

```
int soma = 0; +1
for (int i=0; i<n; i++){ n+1
    for (int j=0; j<n; j++){ n+1
        soma = soma + 1;
    }
}
TOTAL: n^2+2n+2
```



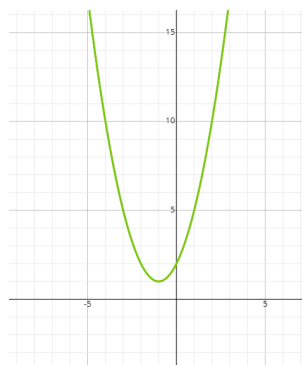
1

f)

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

g)

```
int menor = MAIOR-INTEIRO; +1
for (int i=0; i<n; i++){ n+1
    if (vetor[i] < menor) n+1
        menor = vetor[i];
}
TOTAL: n^2+2n+2
```

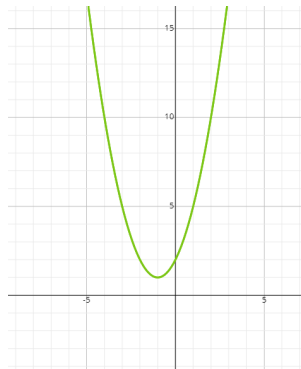


h)

```
int v[][] = new int[n][n]; +1
```

```
for (int i=0; i<n; i++){ n+1
    for (int j=0; j<n; j++){ n+1
        v[i][j] = i * j;
    }
}
```

TOTAL: $n^2 + 2n + 2$



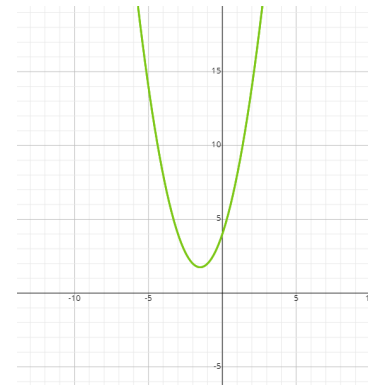
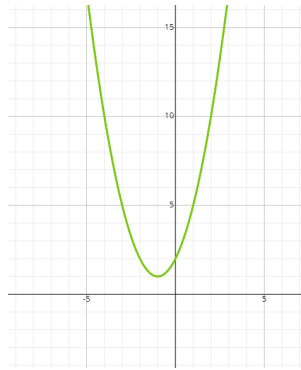
i)

```
int menor = MAIOR-INTEIRO; +1
```

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

TOTAL: melhor caso (menor > 0) $n^2 + 2n + 2$

pior caso (menor < 0): $n^2 + 3n + 4$



j)

```
int menor = MAIOR-INTEIRO; +1
```

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

```
} else if (menor > 0) { +1
```

```
    for (int i=0; i<n*n; i++) n^2+1
```

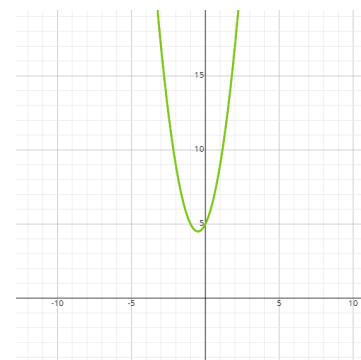
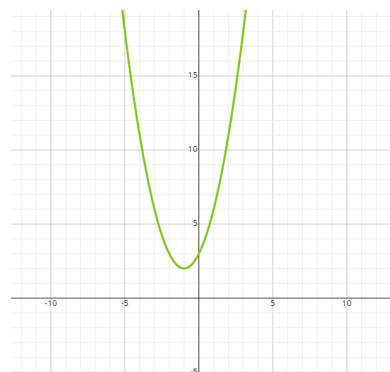
```
        printf("%d\n", menor);
```

```
    } else {
```

```
        printf("%d\n", menor); +1
```

```
    }
```

TOTAL: melhor caso (menor = 0): $n^2 + 2n + 3$ pior caso (menor > 0): $2n^2 + 2n + 5$



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.

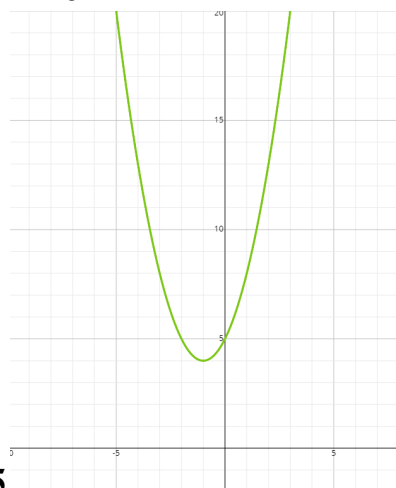
```
Pessoa busca(String nome){
    for (int i = 0; i < pessoas.size(); i++){ n+1
        if (pessoas.get(i).getNome().equals(nome)) n+1
            return pessoas.get(i);
    }
    return null; +1
}
```

TOTAL: melhor caso(primeiro elemento do array = objetivo): 3
pior caso(elemento não encontrado): n^2+2n+2
ou (ultimo elemento) n^2+2n+1

a)

```
void exibir(String nome){
    Pessoa p = busca(nome); +1
    if (p != null){ +1
        p.exibirDados(); +1
    }
    else{
        System.out.println("Pessoa não encontrada"); +1
    }
}
```

TOTAL: melhor caso: $3+1+1+1=6$



pior caso: n^2+2n+5

b)

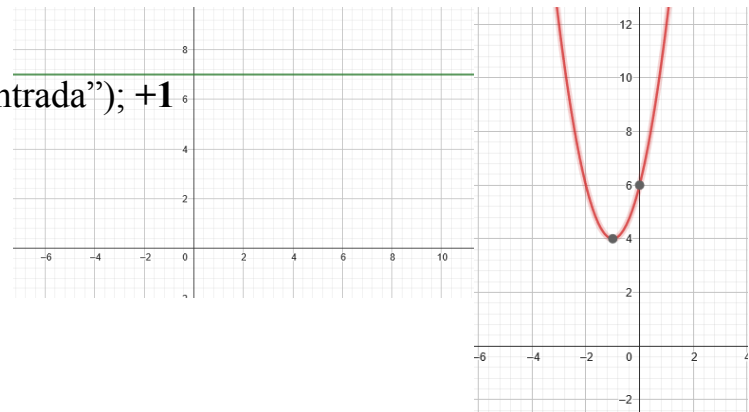
```
void exibir(String nome){
    if (busca(nome) != null){ +2
        busca(nome).exibirDados(); +2
    }
```

```

    }
    else{
        System.out.println("Pessoa não encontrada"); +1
    }
}

```

TOTAL: melhor caso: 7
pior caso: $2(n^2)+4n+6$



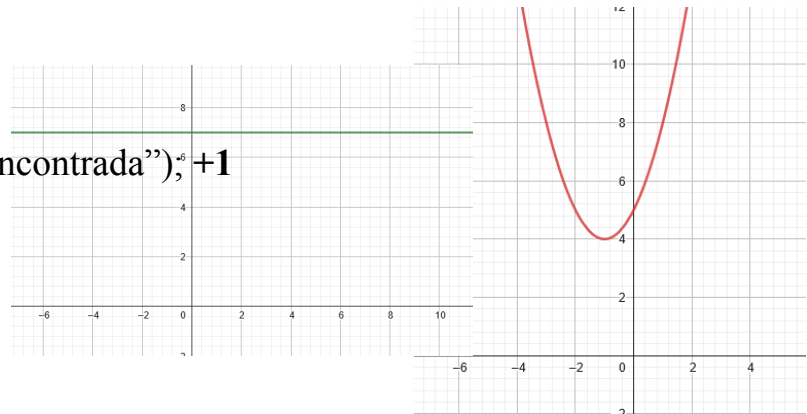
c)

```

void atualizar(String nome, int idade, float salario)
{ Pessoa p = busca(nome); +1
  if (p != null){ +1
    p.setIdade(idade); +1
    p.setSalario(salario); +1
  }
  else{
    System.out.println("Pessoa não encontrada"); +1
  }
}

```

TOTAL: melhor caso: 7
pior caso: n^2+2n+5



d)

```

void atualizar(String nome, int idade, float salario)
{ if (busca(nome) != null){ +1
  busca(nome).setIdade(idade); +1
  busca(nome).setSalario(salario); +1
}
else{
  System.out.println("Pessoa não encontrada");
}
}

```

TOTAL: melhor caso: 12
pior caso: $3(n^2+2n)+6$

