

# ***Alguns Métodos Gerais*** **(GABARITO)**

Unidade I: Análise de Algoritmos

# Exercício (1)

- Faça um método *int somatorioPA(double a, double b, int n)* que retorna o somatório dos  $n$  primeiros termos de uma PA com termo inicial  $a$  e razão  $b$ .

# Exercício (1)

- Faça um método *int somatorioPA(double a, double b, int n)* que retorna o somatório dos *n* primeiros termos de uma PA com termo inicial *a* e razão *b*.

```
int somatorioPA(int a, int b, int n){  
    return ((2*a + b*n)*(n+1))/2;  
}
```

## Exercício (2)

- Um algoritmo de ordenação tradicional é o Inserção. Faça a análise de complexidade desse algoritmo para os números de comparações e movimentações entre registros no pior e melhor caso

```
1:  for (int i = 1; i < n; i++) {  
2:      int tmp = array[i];  
3:      int j = i - 1;  
4:      while ( (j >= 0) && (array[j] > tmp) ){  
5:          array[j + 1] = array[j];  
6:          j--;  
7:      }  
8:      array[j + 1] = tmp;  
9:  }
```

# Exercício (2)

- Um algoritmo de ordenação tradicional é o Inserção. Faça a análise de complexidade desse algoritmo para os números de comparações e movimentações entre registros no pior e melhor caso

```
1:  for (int i = 1; i < n; i++) {
2:      int tmp = array[i];
3:      int j = i - 1;
4:      while ( (j >= 0) && (array[j] > tmp) ){
5:          array[j + 1] = array[j];
6:          j--;
7:      }
8:      array[j + 1] = tmp;
9:  }
```


**Melhor caso:** Elementos em ordem crescente

**Pior caso:** Elementos em ordem decrescente

# Exercício (2): Melhor Caso - Comparações

- Efetuamos uma comparação em cada iteração do laço externo
- Repetimos o laço externo  $(n - 1)$  vezes
- $C(n) = (n - 1) = \Theta(n)$

```
1:  for (int i = 1; i < n; i++) {  
2:      int tmp = array[i];  
3:      int j = i - 1;  
4:      while ( (j >= 0) && (array[j] > tmp) ){  
5:          array[j + 1] = array[j];  
6:          j--;  
7:      }  
8:      array[j + 1] = tmp;  
9:  }
```



# Exercício (2): Pior Caso - Comparações

- Efetuamos  $i$  comparações em cada iteração do laço interno
- Repetimos o laço externo  $(n - 1)$  vezes

- $C(n) = 1 + 2 + 3 + \dots + (n-1) \Rightarrow$   
$$C(n) = \sum_{0 \leq i \leq (n-1)} i = \frac{(n-1)*n}{2} \Rightarrow$$
  
$$C(n) = \Theta(n^2)$$


```
1:  for (int i = 1; i < n; i++) {  
2:      int tmp = array[i];  
3:      int j = i - 1;  
4:      while ( (j >= 0) && (array[j] > tmp) ){  
5:          array[j + 1] = array[j];  
6:          j--;  
7:      }  
8:      array[j + 1] = tmp;  
9:  }
```

# Exercício (2): Número de Movimentações

- Movimentações acontecem nas linhas 2, 5 e 8
- Número de movimentações no laço interno é o de comparações menos um
- Cada iteração do laço externo tem as movimentações do interno mais duas

- $M_i(n) = (C_i(n) - 1) + 2 \Rightarrow$

$$M_i(n) = C_i(n) + 1$$



```
1:  for (int i = 1; i < n; i++) {  
2:      int tmp = array[i];  
3:      int j = i - 1;  
4:      while ( (j >= 0) && (array[j] > tmp) ){  
5:          array[j + 1] = array[j];  
6:          j--;  
7:      }  
8:      array[j + 1] = tmp;  
9:  }
```