Atividade 3 - Análise de Complexidade de Algoritmos

17 de setembro de 2025

Problema A

Procedimento:

```
void imprime(){
    for(int i=0; i<100; i++){
        printf("%d\n", i);
    }
}
```

Resposta: $Comp = 101$

Explicação:
A função realiza um total de 100 comparações bem-sucedidas, ou seja, que repetiram o procedimento dentro do laço, adicionada de mais uma comparação, que retornará falso, que fará com que o fluxo de funcionamento do programa vá para fora do laço de repetição.

Problema B

Procedimento:

```
void imprime(int n){
 for(int i=0; i<n; i++){
  printf("%d\n", i);
      }
}
```

Resposta: $Comp = n + 1$

Explicação:
Similarmente ao procedimento anterior, a função realiza um total de n comparações bem-sucedidas, adicionada de mais uma comparação que retornará falso, totalizando a resposta de $n + 1$.

Problema C

Procedimento:

```
void imprime(int a, int b){
    for(int i=a; i<b; i++){
        printf("%d\n", i);
    }
}
```

Resposta: $Comp = b - a + 1$

Explicação:
O procedimento realiza um total de comparações bem-sucedidas, correspondente ao tamanho do intervalo [a,b), acrescido da comparação final, que acontece quando $a == b$, que determina o fim do laço de repetição.

Problema D

Procedimento:

```
// n = número de linhas
// m = número de colunas
int somaMatriz(int **matriz, int n, int m){
    int soma = 0;
    for(int i = 0; i < n; ++i){
        for(int j = 0; j < m; ++j){
            soma += matriz[i][j];
        }
    }
    return soma;
}
```

Resposta: $Comp = n(m + 1) + n + 1$

Explicação:
O laço mais interno é acionado um total de n vezes e, dentro do laço mais interno, são realizadas um total de $m + 1$ comparações, como um laço de repetição que vimos nos problemas anteriores. Além disso, o laço principal executa um total de $n+1$ comparações, totalizando $n(m + 1) + n + 1$ comparações.

Problema E

Função 1

Procedimento:

```
void f1(int n){
    int i; int j;
    for(i = 0; i < n; i++){
        for(j = n-i; j < n; ++j){
            processa();
        }
    }
```

```
}
```

Resposta: $Comp = n(n+1)/2 + n + 1$

Explicação:

Podemos expandir o laço mais interno do procedimento em função do valor de i e o número de comparações realizadas para cada um deles:

i	comparações
0	1
1	2
2	3
3	4
...	...
n-1	n

O comportamento percebido corresponde exatamente à fórmula da soma dos termos de uma progressão aritmética (PA), tal que $n_{comparaes} = n(n+1)/2$.

Quanto ao laço de repetição principal, ele é executado um total de $n+1$ vezes. Dessa forma, o procedimento realiza um total de $n(n+1)/2 + n + 1$ comparações.

Função 2

Procedimento:

```
void f2(int n){
    int i; int j;
    for(i = n; i >= 0; i--){
        for(j = 1; j < 10; j++){
            processa();
        }
    }
}
```

Resposta: $Comp = 11n + 12$

Explicação:
O loop mais interno sempre realiza um total de 10 comparações, independente do valor de n, e esse loop é acionado um total de $n+1$ vezes, totalizando $10(n+1)$ comparações nesse laço. Já o outro for realiza um total de $n+2$ comparações. Assim, temos:

$$\begin{gathered} 10(n+1) + (n+2) = \\ = 10n + 10 + n + 2 = \\ = 11n + 12 \end{gathered}$$

Função 3

Procedimento:

```
void f3(int n){
    int i;
    for(i = 1; i <= n; i *= 2){
        processa();
    }
}
```

Resposta: $Comp = \lfloor log_2(n) \rfloor + 2$

Explicação:

Podemos analisar a evolução do valor de i durante as iterações do loop:

i
1
2
4
8
16
...
2^{j-1}

O valor de i acompanha o valor das potências de 2. Dessa forma, o loop acontece enquanto $2^{j-1} <= n$, sendo j o número de iterações.

Para encontrar o valor de j, podemos tirar o log_2 em ambos os lados da igualdade:

$$2^{j-1} <= n$$
$$log_2(2^{j-1}) <= log_2(n)$$
$$j - 1 <= log_2(n)$$
$$j <= \lfloor log_2(n) \rfloor + 1$$

Por fim, quando $i > n$, o loop realiza uma comparação final de saída, totalizando $\lfloor log_2(n) \rfloor + 2$.

Problema F

Procedimento:

```
...
char texto[1000];
fgets(texto, 1000, stdin);
int num_a = 0;
for(int i = 0; i < strlen(texto); i++){
  if(texto[i] == 'a'){
    num_a++;
  }
}
...
```

Resposta: $Comp = n^2 + 4n + 2$

Explicação:
O laço de repetição executa um total de $n + 1$ comparações, nas quais chama a função `strlen()`, que também executa um total de $n + 1$ comparações. Dessa forma, são realizadas $(n + 1)(n + 1)$ comparações somente no loop por conta das chamadas da função `strlen()`, mais $n + 1$ comparações com o resultado da função `strlen()`. Além disso, são realizadas n comparações `texto[i] == 'a'` dentro do escopo do laço. Assim, temos:

$$\begin{gathered}(n+1)(n+1)+(n)+(n+1)= \\ =n^2+2n+1+n+n+1= \\ =n^2+4n+2\end{gathered}$$