#### GABRIEL MARTINS PEREIRA - 733875

### 1) Somatórios

a) Progressão Aritmética (PA)

$$S_{n} = \sum_{0 \le i \le n} [a + b.i] = \underbrace{(2a + bn)(n+1)}_{2} \qquad S_{n} = \sum_{0 \le i \le n} [0 + 1.i] = \underbrace{(2.0 + 1.n).(n+1)}_{2} = \underbrace{n.(n+1)}_{2}$$

### b) Propriedades

P1: Combinando Conjuntos

$$\sum_{i \in I} a_i + \sum_{i \in I'} a_i = \sum_{i \in I \cup I'} a_i + \sum_{i \in I \cap I'} a_i$$

P2: Base para a Pertubação

$$S_n + a_{n+1} = a_0 + \sum_{0 \le i \le n} a_{i+1}$$

**c) Progressão Geométrica** → adquirida a partir da base para pertubação

$$S_n = \sum_{0 \le i \le n} a.x^i = \underline{a - a.x}^{n+1}, \text{ para } x \ne 1$$

#### d) Prova por Indução

(Passo Base) → provar que a fórmula é verdadeira para o primeiro valor (Indução) → provar que a fórmula é verdadeira para o primeiro valor

$$S_n = S_{n-1} + a_n$$

$$S_{n-1} = \text{\'e a equação substituindo n por (n-1)} \quad S_n = \frac{2n^3 + 3n^2 + n}{6}$$

$$a_n = \text{n-\'esimo termo da sequência} \quad S_n = \frac{n \cdot (n+1)/(2n+1)}{6}$$

## 2) Análise de Algoritmos

```
1) f(n) = O(f(n))

2) c x O(f(n)) = O(f(n))

3) O(f(n)) + O(f(n)) = O(f(n))

4) O(O(f(n))) = O(f(n))

5) O(f(n)) + O(g(n)) = O(máximo(f(n),g(n)))

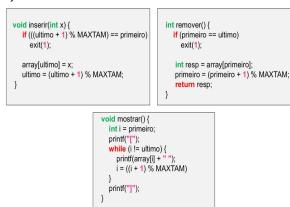
6) O(f(n)) x O(g(n)) = O(f(n) x g(n))

7) f(n) x O(g(n)) = O(f(n) x g(n))
```

\*) As mesmas propriedades são aplicadas para  $\Omega$  e  $\Theta$ 

# 3) Estrutura de Dados Lineares

#### a) Fila Flexível



```
int removerInicio() {
                                                                int remover(int pos) {
   if (n == 0)
     exit(1);
                                                                 if (n == 0 || pos < 0 || pos >= n)
                                                                    exit(1);
   int resp = array[0];
                                                                  int resp = array[pos];
   for (int i = 0; i < n; i++){
     array[i] = array[i+1];
                                                                  for (int i = pos; i < n; i++){
                                                                   array[i] = array[i+1];
                                int removerFim() {
   return resp;
                                  if(n == 0)
                                    exit(1);
                                                                 return resp:
                                  return array[-n];
void inserirInicio(int x) {
                                                               void inserir(int x, int pos) {
 if (n >= MAXTAM)
                                                               if (n \ge MAXTAM || pos < 0 || pos > n)
   exit(1);
 //levar elementos para o fim do array for (int i = n; i > 0; i--){
                                                                //levar elementos para o fim do array
                                                                for (int i = n; i > pos; i--){
   array[i] = array[i-1];
                                                                  array[i] = array[i-1];
 array[0] = x;
                                                                array[pos] = x;
                               void inserirFim(int x) {
                                if (n \ge MAXTAM)
                                  exit(1);
                                array[n] = x;
```