**MARCOS ANTONIO LOMMEZ CANDIDO RIBEIRO**

• Quando existir um laço começando em 0 ate N teremos, N repetições, por outro lado quando começar em ‘a’ teremos n-a repetições

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

for (int i = a; i<n; i++) = n-a

• O laco do algoritmo de seleção não precisa ir até a posição n-1 (ultima), por isso seu limite superior será (n-2), enquanto o laco interno realiza i<n operações, o que pode ser expresso com a equação (n-i-1) onde (n-i) simula um decremento no valor de ‘i’. Logo sua formula será:

<< associatividade

<< substituição pela formula (n\*n)

<< substituição pelo somatório de Gauss

• Também podemos manipular somas através dos seus conjuntos

∑ai + ∑ai = ∑ai + ∑ai

i ∈ I i ∈ I' i ∈ I ⋃ I' i ∈ I ⋂ I'

• E através da base para perturbação

Sn + an+1 = a0 + ∑ai+1 onde ai = a.xi

• Exemplo aplicado a formula do somatório da PG

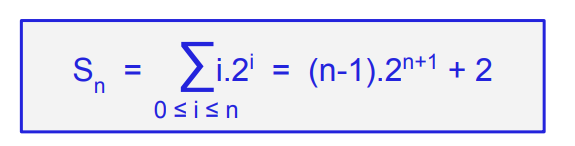
Sn + a.xn+1= a.x0 + ∑ a.xi+1 = Sn + a.xn+1= a.x0 + ∑ a.xi.x

Sn + a.xn+1= a.x0 + x.∑ a.xi = Sn + a.xn+1= a.x0 + x. Sn

Sn + a.xn+1= a + x. Sn = Sn - x. Sn= a - a.xn+1

(1-x)Sn = a - a.xn+1  = Sn = (para i diferente de 1)

• Exemplo de Somatório encontrado através da aplicação da formula acima junto da base para perturbação:



• Outras formulas uteis:

= n(n+1)

= n+1

=

ai = i2

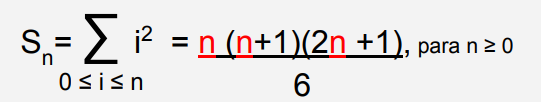
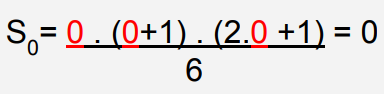
• Uma formula de somatório pode ser provada correta utilizando a indução.

A mesma é dividida em duas partes, sendo

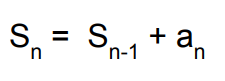
* Passo base: Substituir n pelo primeiro valor e provar seu resultado
* Indução propriamente dita: A formula devera ser valida ao substituir n por n-1

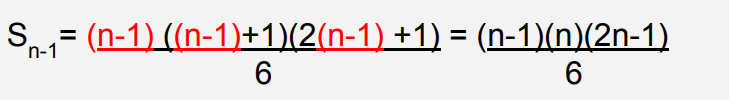
Exemplos:

Passo base

Indução propriamente dita:





• Para calcular o numero de repetições de um laço do tipo logarítmico (n/=2) realizaremos

, o +1 ocorre por causa que o logaritmo não conta o valor do laço onde n será igual a 1.

• A pesquisa sequencial possui custo Θ(n) e a pesquisa binaria possui custo Θ(lg(n))

O custo mínimo da ordenação é Θ(n.lg(n)), logo para um array não ordenado deveremos usar a pesquisa sequencial, a não ser sejam realizados n.lg2 (n) pesquisas, a partir dai vale mais apena ordenar e pesquisar binariamente.

A definição das notações nasce do conceito de que:

Para uma função g(n) ser O(f(n)), ela devera possuir constantes ‘c’ e ‘m’ para que sua “reta” ultrapasse em determinado momento (m) a “reta” de (g).

Ao utilizar a notação estes valores ficarão implícitos, é necessário apenas que eles existam para que a premissa se torne verdadeira. Pelo mesmo lado o mesmo é necessário para Ω(f(n)), a diferença é que a “reta” não pode ultrapassar

E por fim para Θ(f(n)) estes valores deverão existir onde seja possível que a “reta” esteja entre O e Ω de forma justa.

• Exemplo de notação de complexidade

