Bài 1: Giả sử thủ tục A đòi hỏi M đơn vị thời gian tính toán để thực thi và M là hằng số. Hãy phân tích độ phức tạp C(n) của giải thuật sau đây với n là kích thước dữ liệu nhập và b là một số nguyên dương lớn hơn 1:

J:=1;

While j<=n do

Begin

Call A;

J:=j\*b

End;

Bài 2: Giả sử thủ tục A đòi hỏi M đơn vị thời gian tính toán để thực thi và M là hằng số. Hãy phân tích độ phức tạp C(n) của giải thuật sau đây với n là kích thước của dữ liệu nhập:

For i:=1 to n do

For j:=1 to i do

For k:=1 to j do

Modul A;

Bài 3: Hãy phân tích độ phức tạp C(n) của giải thuật sau đây với n là kích thước dữ liệu nhập:

Function mystery(n)

R:=0;

For i:=1 to n-1 do

For j:=i+1 to n do

For k:=1 to j do

r:=r+1

return(r)

**solution:**

Function mystery(n)

r:=0;

For i:=1 to n-1 do **// n-1 lần**

For j:=i+1 to n do **// (n-1) + (n-2)+(n-3)+… 3+2+1= [n(n+1)/2] – n= N lần**

For k:=1 to j do **// j=2 -> n =>S2= 2+3+4+..+n =[n(n+1)/2] -1**

**// j=3-> n => S3 =3+4+..+n=[n(n+1)/2] -1-2**

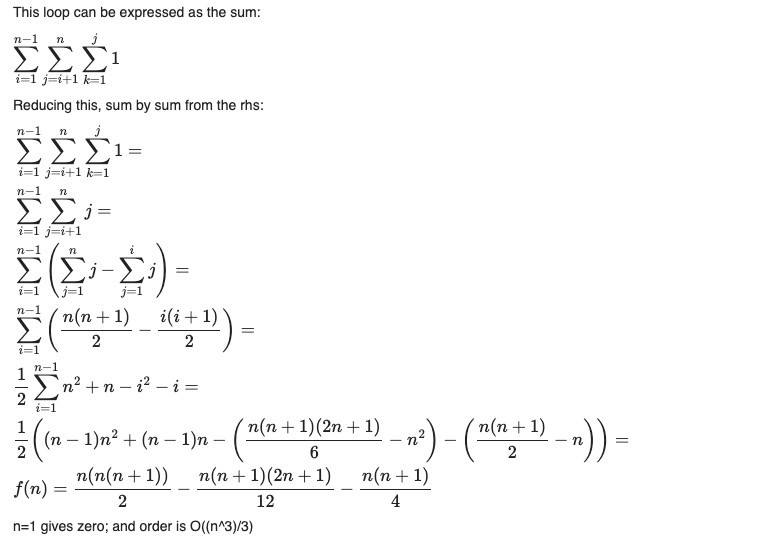
**….**

**// j=n -> n => Sn = n**

r:=r+1

return(r) **T(n)=N(N+1)/2 +1=>O()**

1+2+3+….+(n-2)+(n-1)+n=n\*(n+1)/2



Bài 4:

xác định độ phức tạp

Int sum(int n){

If(n==1) return 1

Else return sum(n-1)+1;

}

Bài 5: cho một chương trình đệ quy với hệ thức truy hồi sau:

Với N là lũy thừa của 2

Hãy hệ thức truy hồi nêu trên.