**Bài 202:** T(n) =

Ta có: T(n) = 3T(n-1) +2

=3(3T(n-2) +2) +2

= 32T(n-2) + 3.2+2

=32(3T(n-3) +2) +3.2+2

= 33T(n-3) + 32.2+3.2+2

=……..

=3n-1T(n-(n-1)) + 2(3n-1 + 3n-2+…+30)

= 3n-1 + 2.

= 3n-1 + 3n -1

=O(3n)

**Bài 203:** T(n) =

* Ta có: T (n) = 3T(n-1) -15

= 3(3T(n-2) -15) -15

=32T(n-2) – 3.15 -15

=32(3T(n-3) -15) -3.15-15

=33T(n-3) – 32.15 – 3.15 -15

=….

=3n-1T(n-(n-1)) – 15( 3n-2 + 3n-3 + … + 30)

=3n-1.8 – 15.

= +

=O(3n)

**Bài 204 :** T(n) =

Ta có: T(n) =T(n-1) + n - 1

= T(n-2) + n-2 + n-1

=T(n-3) + n-3 +n-2 + n-1

= T(n-3) +3.n –(1+2+3)

=…

=T(n-(n-1)) + n.(n-1) –((n-1)+….2+1)

=2 +n(n-1) –

= O(n2)

**Bài 205:** T(n) =

Ta có: T(n) =T(n-1) +2n -3

=T(n-2) +2(n-1) -3+2n-3

=T(n-3) + 2(n-2) -3+2(n-1) -3 +2n-3

=T(n-3) +3.(2n) -3.3 –(22 +2)

=…

=T(n-(n-1)) +(n-1).2n –n.3 – (2n-2 + …+2)

=T(1) +2n(n-1) -3.n -

=3+2.n2 –5n -2n-1+2

=O(2n)

**Bài 206:** T(n) =

Ta có:

T(n) = 2T(n – 1) + n – 1

= 2[2T(n – 2) + n – 1 – 1] + n – 1

= …

= 2n-1T(1) + 2n-2.1 + 2n-3.2 +…+ 21.(n-2) + 20(n – 1)

= 2n-1 + [ 2n-2.2 + 2n-3.3 +…+ ( n – 2)] – [ 2n-2 + 2n-3 +…+ 2 + 20]

= 2n-1 – (2n-1 – 1) + [ 2n-2.2 + 2n-3.3 +…+ ( n – 2)]

= O (2n).

**Bài 207**: T(n) =

Ta có: T(n) = 2T(n-1) + 3n + 1

= 2[2T(n-2) +3(n-1) +1] + 3n + 1

= 22T(n-2) + 2.3(n -1)+2 +3n+1

=22[2T(n-3) + 3(n-2) +1] + 2.3(n -1)+2 +3n+1

= 23T(n-3) + 22.3(n-2) +22.1 + 2.3(n -1)+2 +3n+1

=…

= 2n-1T(n-(n-1)) +[ 3.2.2n-2+ … + 2.3(n-1) + 3n] +(2n-2+ … + 2+1)

=5.2n-1 +(2n-1 -1) +[ 3.2.2n-2+ … + 2.3(n-1) + 3n]

=O(2n)

**Bài 208:** T(1) = 1 n=1

T(n) = 2T() + 6n - 1 n>1,

Ta có:

T(n) = 2T( )+ 6n – 1

= 2[2T() +6. -1] +6n -1

= 22T() +6n -2 + 6n -1

=22[2T() + 6. -1] + 6n + 6n -2 -1

= 23T() + 3.6n – 22 -2 -1

=……………………………………………

=2kT() + k.6n – 2k-1 - … - 2 – 1 (1)

Đặt n = n

1. = n.T(1) + 6n.logn – ()

= n.1 + 6n.logn + 1 - n

= 6n.logn +1

Vậy T(n) = O(nlogn)

**Bài 209 :** T(n) =

Ta có: T(n) = 2T(n/2) + 3n +2

= 2(2T(n/22) + 3.n/2 +2) +3n +2

= 22T(n/22) + 3n + 22 + 3n +2

= 23T(n/23) + 3n + 23+3n+22 +3n+2

=…

=2kT(n/2k) + k.3n +(2+…+2k)

=2kT(n/2k) + k.3n + 2.

Đặt: 2k = n => k = logn

* T(n) = n.4 + 3n.logn + 2.2.n =O(nlogn)

**Bài 210 :** T(n) =

Ta có: T(n) =6T() +2n +3

=6[6T() +2() +3] + 2n + 3

=62[6T() +2. ()+3] +6. 2() +6.3 + 2n + 3

=T() + 2n + 2n + 2n + .3 + 6.3 + 3

=………………………………

= T() + 2n.k + 3.(+ …….+ 1)

Với k= log6n:

T(n)=n.T(1) +2n. log6n + 3.()

=n + 2n. log6n - (1 – n)

= O(n. log6n)

**Bài 211:** T(n) =

Ta có: T(n) =6T() +3n -1

=6[6T() +3() -1] +3n -1

=62[6T() +3. ()-1] +6. 3() - 6 + 3n -1

=T() + 3n + 3n + 3n - - 6 - 1

=………………………….

=T() + 3n.k - (+ …….+ 1)

Với k=log6n:

T(n)=nT(1) +3. log6n.n +

=n.3 + 3. log6n.n –

=O(n. log6n)

**Bài 212** : T(n) =

Ta có: T(n) = 4T(n/3 )+2n-1

=4[4(n/32) +2(n/3)-1] )+2n-1

=42[4(n/33) +2(n/32)-1] +4. 2(n/3) -4+2n-1

=…

=4kT(n/3k) +2n(4k-1/3k-1+…+1) –(4k-1+…+1)

=4kT(n/3k) +2n[((4k/3k) -1)/(4/3 -1)] – (4k-1)/(4-1)

Đặt : 3k =n => k= log3n:

T(n)= 4log3n .3 +2n.3(4log3n /n)-1 –( 4log3n -1)/3

=O(4log3n)

**Bài 213:** T(n) =

Ta có: T(n) =4T(n/3 )+3n-5

=4[4T(n/32) +3(n/3)-5] +3n-5

=42[4T(n/33) +3(n/32)-5] +4n -4.5 +3n-5

=…

=4kT(n/3k) +3n(4k-1/3k-1 +…+4/3+1) - 5(4k-1+…+4+1)

=4kT(n/3k) +3n.[(4k/3k -1)/(4/3-1)] -5.(4k-1)/(4-1)

=

Đặt: n=3k => k=log3n:

T(n)= 4log3n .2 +3n.(4log3n/n -1).3 -5.(4log3n -1)/3

= O(4log3n)

**Bài 214 :** T(n) =

Đặt n =

T(n) = T() = 3T + – 2m

= 3(3T() +22(m-1)  - 2(m -1)) + 22m – 2m

= 32 .T(2m-2) + 3.22(m-1) - 3.2(m-1) + 22m - 2m

=……

= 3m T(1) + 3m-1 .22 + .. + 3.22(m-1) + 22m - 2.3m-1 -.. - 3.2m-1 – 2m

= 3m  + 3m () – 3m( + + .. +(m)

= 3m  + 3m .. - 3m . ( )

= 3m + 4.4m – 4.3m – + 2.2m

= 4(m + 1) - 5. 3m + 2(m + 1) = O(4m) = O(n2)

**Bài 215:** T(n) =

Đặt n =

T(n) = T() = 3T + – 2m + 1

= 3(3 T() +2 2(m-1)  - 2(m -1) + 1) + 22m – 2m +1

= 32 .T(2m-2) + 3.22(m-1) - 3.2(m-1) + 3 + 22m - 2m + 1

=……

= 3m T(1) + (3m-1 .22 +…+ 30.22m)- (21.3m-1 +…+ 2m.30 )+ (30 + …+ 3m-1)

= 3m  + 3m () – 3m( + + .. +(m) +

= 3m  + 3m .. -3m . ( ) +

= 3m + 4.4m – 4.3m – + 2.2m  + -

= 4(m + 1) - . 3m + 2(m + 1) - = O(4m) = O(n2)

**Bài 216 :** T(n) =

Ta có: T(n) = 3T(n/2) + n-2

= 3(3T(n/22) + n/2 -2) +n-2

=32 T(n/22) + 3.n/2 – 2.3 +n-2

=33 T(n/23) + 32.n/22 -32.2 +3.n/2 – 2.3 +n-2

=33 T(n/23) +n((3/2)2 + 3/2+1) -2(32 +3+1)

=…

=3kT(n/2k) + n((3/2)k-1+…+ 3/2+1) – 2(3k-1+…+ 3+1)

=3kT(n/2k) +n. - 2.

Đặt: 2k =n => k =logn

* T(n) =3logn + 2n( -3logn +1

=2.3logn -2n +1

= O(3logn)

**Bài 217:**  T(n) =

T(n)= 4T(n/3) +n2

Đặt n=3m

T(n)= 4 T(3m-1) +32m = 42 T(3m-2) +4.32(m-1) +

=42 T(3m-2) +4/32. 32m +32m

=…

=4m T(3m-m) +(4/32)m-1. 32m +4/3. 32m +32m

=4m +1/(1-4/32) 32m = O(32m )

= O(n2)

**Bài 218:** T(n) =

Đặt n=3m

T(n) = 4 T(3m-1) +32m = 42 T(3m-2) +4.32(m-1) + 32m

=42 T(3m-2) +4/32. 32m + 32m

=…

=4m T(3m-m) +(4/32)m-1. 32m +…+4/32. 32m +32m

= 4m + . 32(m-1)

O(32m) = O(n2).

**Bài 219:** T(n) =

T(n)= 4T() +n2 -7n +5

=4[4T() + –+ 5] + n2 -7n +5

=42T() + 4.– 4. + 4.5 + n2 -7n +5

=…

= 4kT() + n2 .((+………….+ 1) -7n. .((+…+ 1) + 5(

Đặt n=3k, k= log3n

T(n) = T(1) +– 7n + 5

=>T(n)=O(n2)

**Bài 220:**

Đặt n = 4m = 22m => m = . logn

=> T(n) = T(4m) = T(4m-1) + 2m + 1

= T(4m-2) + 2m-1 + 1 + 2m + 1

= …

= T(1) + [ 21 + … + 2m ] + m

= 20 + 21 + … + 2m + m

= + m = 2m+1 + m – 1

= 2. + . logn – 1 = O( )

**Bài 221:**

Không mất tính tổng quát,giả sử : 0<α<β<1

Ta c/m quy nạp công thức sau:

T(n)=∑cim .T(αi.βm-i.n) +mcn : sao cho 1< αm.n

Với m=1 thì: T(n)=∑ci1 .T(αi.β1-i.n) +cn (\*)

= c01 .T(α0.β1.n) + c11 .T(α1.β0.n) +cn

=T(βn) +T(αn) +cn => (đúng)

Giả sử CT (\*) đúng đến m,ta chứng minh nó đúng với m+1:

T(n) =∑cim .T(αi.βm-i.n) +mcn

=∑cim .T(αi+1.βm-i.n) +∑cim T(αi.βm+1-i.n)+ ∑cim c αi.βm-i.n +mcn

=∑i=0m-1cim .T(αi+1.βm-i.n) + cmm .T(αm+1.β0.n) + c0m.T(α0.βm+1.n) + ∑cim.T(αi.βm+1-i.n) +c(∑cim .αi.βm-i).n +mcn

=∑i=0m cim .T(αi+1.βm+1-i-1.n)+ cm+1m+1 .T(αm+1.n) + c0m+1.T(βm+1.n)+ ∑cim.T(αi.βm+1-i.n)+c(α+β)m.n +mcn

=∑i=1mci-1m .T(αi.βm+1-i.n) + cm+1m+1 .T(αm+1.n) + c0m+1.T(βm+1.n)+ ∑cim.T(αi.βm+1i.n)+ cn +mcn

=∑i=1m( ci-1m + cim) T(αi.βm+1-i.n)  + cm+1m+1 .T(αm+1.n) +c0m+1.T(βm+1.n)+(m+1)cn

=∑i=1mcim+1 .T(αi.βm+1-i.n)+ cm+1m+1 .T(αm+1.n) + c0m+1.T(βm+1.n)+(m+1)cn

=∑i=0m+1cim+1 .T(αi.βm+1-i.n)+(m+1)cn

Vậy (\*) đúng với mọi m. CT được thực hiện cho đến khi:

αi.βm-i.n=1,nó thực hiện tối đa số lần mà:

βm.n =1⬄(1/β)m=n⬄m=log1/β n và tối thiểu số lần:

(1/α)m =n⬄m= log1/α n khi mà: αi.βm-i.n < = 1thì:

T(n)=∑cim .T(1) +mcn =∑i=0m +mcn =2m + mcn

=>T(n) = O(nlogn)

**Bài 222:**

Ta c/m quy nạp theo n:

1.Với n=1 =>đúng.vì:(Ф-Ψ)/√5= (1+√5-1 + √5)/(2. √5)=1 =F(1)

2.Giả sử đúng đến n=k tức là:

F(k) =(Фk-Ψk)/√5

3.Ta phải chứng minh nó đúng với n=k+1 tức là c/m:

F(k+1) =(Фk+1-Ψk+1)/√5.Thật vậy có:

F(k+1) =F(k) +F(k-1) =(Фk-Ψk)/√5 + (Фk-1-Ψk-1)/√5

=[( (1+√5)/2 )k-1.(( 1+√5)/2) +1) -( (1-√5)/2 )k-1 .(( 1-√5)/2) +1)]/ √5

=[( (1+√5)/2 )k-1.(3+√5)/4) -( (1-√5)/2 )k-1.( 3+√5)/4)]/ √5

=[(1+√5)/2 )k-1 - (1-√5)/2 )k-1]/ √5

=(Фk+1-Ψk+1)/√5 (=>đpcm)

**Bài 231**: T(n) =

Ta có: T(n) = aT(n/c) +b

=a[a.T(n/c2) + b] +b

=a2T(n/c2) +ab +b

= a2[a.T(n/c3) + b] +ab +b

= a3 .T(n/c3) +a2b + ab +b

=…..

=ak. T(n/ck ) +b(ak-1 +… +a+1)

= ak. T(n/ck ) +b.

Đặt: n=ck => k = logcn

=>T(n) =alogcn.d +[(1- alogcn)/(1-a)].b

=O(alogcn)

**Bài 232**: T(n) =

Ta có : T(n) = aT(n/c) +bn2

= a[a.T(n/c2) + b(n/c)2] +bn2

= a2T(n/c2) +ab(n/c)2 +bn2

= a2[a.T(n/c3) + b(n/c2)2] +ab(n/c)2 +bn2

= a3 .T(n/c3) +a2b(n/c2)2 +ab(n/c)2 +bn2

=……

=akT(n/ck) +bn2(

=ak.T(n/ck) + bn2.

Đặt : n= ck =>k= logcn

=>T(n)= d.alogcn +bn2.[( alogcn)/n2 - 1)/(a/c2 -1)]

=O(n2. alogcn)

**Bài II:**

**Bài 1** *. In biểu diễn nhị phân của số nguyên*

Ta chỉ quan tâm số phép thực hiện của chương trình đệ quy

* he2(long n) . Ký hiệu là T1(n)

Ta thấy :

(n) =  () + 1  (n) = log n + 1

 (1) = 1

(n) = O() 

**Bài 2 :** *Phân tích số nguyên thành tích các thừa số nguyên tố*

Ta đánh giá độ phức tạp của thuật toán là : (n) = O(n)

**Bài 3 :** *Tìm số Fibonacci thứ n*

Ta chỉ tính số lệnh được thực hiện trong vòng lặp đệ quy .Tính sô Fibonacci thứ n và đặt nó là (n)

Ta có : (n) = 2 + (n-1) + (n-2)

(0) =  (1) = 1

Đặt U(n) =  (n) + 2

U(n) = U(n-1) + U(n-2) +

Dãy U(n) có dạng tổng quát : U(n) = α λ + β λ

Trong đó λ, λ là nghiệm của phương trình λ = λ +1 λ = 

λ = 

Ta thấy : | λ| < 1 , | λ| > 1 nên

U(n) = O(λ) (n) = O(λ)

**Bài 4** : *Tháp Hà Nội*

Gọi (n) là số lệnh thực hiện trong ThapHN(n)

Ta có công thức truy hồi : (n) = 1 + 2(n-1)

(1) = 1

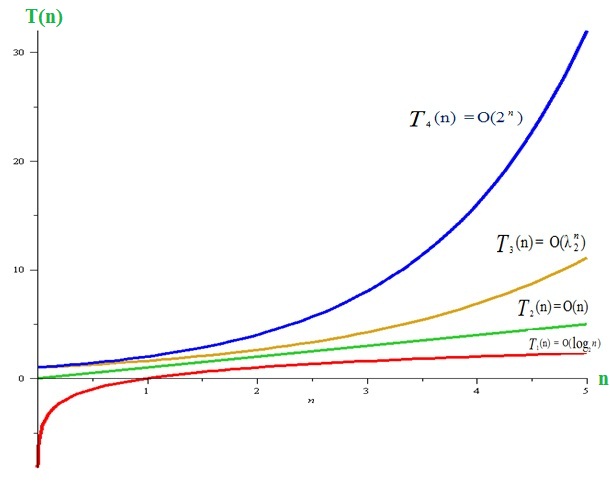
(n) = 2 - 1 = O(2)

* Đưa ra đánh giá ( có thể bằng bảng hoặc biểu đồ ) sự tăng trưởng của thời gian thực hiện chương trình theo kích thước dữ liệu vào .

Sau khi đánh giá thời gian thực hiện thuật toán của các bài toán đệ quy trên .Ta có

* + (n) = O()  ( Biểu diễn nhị phân của số nguyên )
  + (n) = O(n) ( Phân tích số nguyên thành tích các thừa số nguyên tố )
  + (n) = O(λ) ( Tìm số Fibonacci thứ n )
  + (n) = O(2) ( Tháp Hà Nội )

**Biểu đồ sự tăng trưởng của thời gian thực hiện chương trình**



**Bài III:**

Bài 3: Đưa ra đánh giá (có thể bằng bảng hoặc bằng biểu đồ) sự tăng trưởng của thời gian thực hiện chương trình theo kích thước dữ liệu vào.

1, Tìm kiếm phần tử trên mảng được sắp:

* + Độ phức tạp thuật toán là:

T(n) =

⇒ T(*n*) = O(log *n*)

2, Tìm max, min của dãy số :

* + Độ phức tạp thuật toán

T(n) =

⇒ T(*n*) = O(*n*)

3, Tìm cặp điểm gần nhất trên mặt phẳng.

Cách 1: Bai3\_32 : Ta duyệt tất cả các điểm trong N điểm cho trước, so sánh khoảng cách giữa các cặp điểm đó rồi chọn ra khoảng cách ngắn nhất.

* Thuật toán này có độ phức tạp là: O(n2)

Cách 2: Bai3\_31 : Ta thực hiện thuật toán chia để trị:

* Độ phức tạp của thuật toán là : O(nlogn)

=>Ta đánh giá sự tăng trưởng của thời gian thực hiện các chương trình trên bằng biểu đồ sau:



**Biểu đồ sự tăng trưởng của thời gian thực hiện chương trình**

**Bài IV:**

***Bài toán chia thưởng****:* Có *m* vật thưởng được chia cho *n* học sinh giỏi có xếp hạng theo thứ tự từ 1 đến *n*. Hỏi có bao nhiêu cách chia các phần thưởng thoả mãn các điều kiện sau:

1. Học sinh giỏi hơn có số thưởng không ít hơn bạn kém hơn;
2. (ii) *m* vật thưởng phải chia hết cho các học sinh.

**Phân tích bài toán:**

Gọi chia (m,n) là số cách chia m phần thưởng cho n học sinh, ta thấy:

1. Nếu k có học sinh nào thì(n = 0) thì không có cách chia nào (chia = 0).
2. Nếu không có phần thưởng nào (m = 0) thì có 1 cách chia (chia = 1, mỗi học sinh nhận 0 phần thưởng). Ta cũng quy ước chia(0,0) = 1.
3. Nếu số phần thưởng ít hơn số học sinh (m < n) thì từ học sinh thứ m +1 trở đi sẽ không được nhận phần thưởng nào, tức là phần thưởng chỉ có thể chia tối đa cho m học sinh: chia(m,n) = chia(m,n), nếu m < n.
4. Ta xét trường hợp m >= n. Ta tách các phương án chia thành 2 nhóm k giao nhau:

* Nhóm thứ 1 gồm các phương án trong đó học sinh thứ n không được nhận thưởng, tức là m phần thưởng chỉ chia cho n – 1 học sinh và do đó số cách chia, số phần tử của nhóm này là: chia(m,n-1).
* Nhóm thứ 2 gồm những phương án mà học sinh thứ n cũng được nhận thưởng. Khi đó, do học sinh đứng cuối bảng thành tích được nhận phần thưởng thì mọi học sinh khác cũng có phần thưởng. Do ai cũng có phần thưởng lên ta bớt của mỗi người 1 phần thưởng(để họ lĩnh sau), số phần thưởng còn lại (m-n) xẽ được chia cho n học sinh. Số cách chia khi đó xẽ là chia(m – n,n).

Tổng số cách chia cho trường hợp m >= n sẽ là tổng số phần tử của hai nhóm, ta có chia(m,n) = chia(m,n – 1) + chia(m – n,n).

**Xây dựng giải thuật:**

**Function chia(m,n: integer) :longint;**

**Begin**

**1. if m = 0 then**

**2. chia:=1**

**else// {m > 0}**

**3. if n = 0 then //{m > 0;n = 0}**

**4. chia:=0**

**else {m,n > 0}**

**5. if m < n then// {0 < m < n}**

**6. chia:= chia(m,m)**

**else //{m >= n >0}**

**7. chia:= chia(m –n,n) + chia(m,n – 1);**

**End;**

**Đánh gía độ phức tạp của thuật toán:**

Chia : bước 1 và 3 và 5 : (1)

Trị : bước 2 và 4 và 6 : ()

Hợp lại : bước 7 : (n)

Tổng kết:

T(n) = nếu n =0

) + 1 nếu n > 0

Vậy độ phức tạp của thuật toán T(n) = O(n)

***Bài toán xếp balo***: Có *n* đồ vật, mỗi vật có trọng lượng *Pi* và giá trị *Vi* (*i*=1..*n*). Có một chiếc balo có thể chứa trọng lượng tối đa là *M*. Hãy xác định tổng giá trị lớn nhất của các vật có thể đưa vào balo. Chỉ ra một cách cho các vật vào balo.

Phân tích bài toán:

Giả sử chọn đồ vật i(trong mức giới hạn trọng lượng), cập nhật tổng trọng lượng và giá trị.

Lặp tiếp quá trình chọn tới khi không thể thêm được nữa kiểm tra xem cách này có tốt hơn không.

Bỏ chọn từng đồ vật 1, mỗi bước bỏ lại rẽ nhánh ra theo 1 cách chọn khác.

Do lúc nào cũng phải bắt đầu chọn từ 1 vật nên ta cho thêm 1 vật có giá trị và trọng lượng = 0 để bắt đầu mà không sợ ảnh hưởng tới kết quả.

**Xây dựng thuật giải**

Giatrimax = 0;

Select(0,0,0);

Select(chiso,trongluong,giatri)

dd[chiso] = true;

//danh dau chon do vat thu chiso

tongtrongluong += trongluong[chiso];

tonggiatri += giatri[chiso];

//tinh lai tongtrongluong va tonggiatri

for 1 to n do

//duyet tat ca cac do vat duoc chon

If(dd[i] == false && tongtrongluong + trongluong[i] <= MAX\_WEIGHT)

select(i,tongtrongluong,tonggiatri);

//het vong lap cung la khi khong the them do vat duoc nua, day la 1 cach.

//neu cach nay tot hon thi luu lai cach nay

if(tonggiatri > giatrimax)

for 1 to n do

cuoi[i] = dd[i];

//do khong the them duoc nua len ta thu bo do vat nay di va theo huong khac

dd[i] = false;

}

**Đánh giá thuật toán**

Do vét cạn tất cả các trường hợp có thể nên số phép tính toán rất lớn

Độ phức tạp O(n!).