***Bài 1:***

1. ***Bài 202-220 (trang 37,38) sách Ian Parberry - Problems on Algorithms.pdf***
2. **T(1) = 1, and for all n 2, T(n) = 3T(n – 1) + 2**

T(n) = 3.T(n – 1) + 2

= 3[3T(n – 2) + 2] + 2

= .T(n – 2) + 3.2 + 2

= … =

=

=

Quá trình kết thúc n = k + 1

T(n) =

1. **T(1) = 8, and for all n 2, T(n) = 3T(n – 1) – 15**

=

=

=

Quá trình kết thúc n = k + 1

=

1. **T(1) = 2, and for all n 2, T(n) = T(n – 1) + n – 1**

=

=

=

=

Quá trình kết thúc n = k + 1

= 2+

1. **T(1) = 3, and for all n 2, T(n) = 2T(n – 1) + 2n – 3**

=

=

=

=

=

Quá trình kết thúc n = k + 1

=

1. **T(1) = 1, and for all n 2, T(n) = 2T(n – 1) + n – 1**

Đặt

=

=

=

=

=

=

Quá trình kết thúc: n = k + 1

= 2.

=

=

1. **T(1) = 5, and for all n 2, T(n) = 2T(n – 1) + 3n + 1**

Đặt

=

=

=

=

Quá trình kết thúc n = k + 1

=

=

=

1. **T(1) = 1, and for all n 2 a power of 2, T(n) = 2T(n/2) + 6n – 1**

=

=

=

=

Quá trình kết thúc

=

=

1. **T(1) = 4, and for all n 2 a power of 2, T(n) = 2T(n/2) + 3n + 2**

=

=

=

Quá trình kết thúc

=

=

=

1. **T(1) = 1, and for all n 2 a power of 6, T(n) = 6T(n/6) + 2n + 3**

=

=

=

Quá trình kết thúc

1. **T(1) = 3, and for all n 2 a power of 6, T(n) = 6T(n/6) + 3n – 1**

=

=

=

=

=

Quá trình kết thúc

=

1. **T(1) = 3, and for all n 2 a power of 3, T(n) = 4T(n/3) + 2n – 1**

=

=

=

=

Quá trình kết thúckhi

=

1. **T(1) = 1, and for all n 2 a power of 3, T(n) = 4T(n/3) + 3n – 5**

=

Quá trình kết thúckhi

=

1. **T(1) = 1, and for all n 2 a power of 2, T(n) = 3T(n/2) + n**

Quá trình kết thúckhi

Giải:

1. **T(1) = 1, and for all n 2 a power of 2, T(n) = 3T(n/2) + n – 2**

Quá trình kết thúckhi

1. **T(1) = 1, and for all n 2 a power of 2,T(n) = 3T(n/2) + 5n – 7**

T(n) =

=

=

=

=

Quá trình kết thúckhi n =

T(n) =

= …

1. **T(1) = 1, and for all n 2 a power of 3, T(n) = 4T(n/3) +**

=

=

Quá trình kết thúc

=

=

1. **T(1) = 1, and for all n 2 a power of 3, T(n) = 4T(n/3) +**

=

=

Quá trình kết thúc

=

=

1. **T(1) = 1, and for all n 4 a power of 4, T(n) = T(n/4) +**

=

=

Quá trình kết thúc

=

=

=

=

1. ***Bài 221-222 (trang 38) sách Ian Parberry - Problems on Algorithms.pdf***
2. **.Chứng minh rằng**

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

Ta chứng minh quy nạp công thức sau:

T(n) = . T() + m.c.n (\*)

Sao cho: 1 <

Với m = 1 thì:

T(n) = .T() + c.n

= .T() + .T() + c.n

= T(β.n) + T(α.n) + c.n (đúng)

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

T(n) = T() + m.c.n

= [T() + T() + c.] + m.c.n

= T() + T() + .c. + m.c.n

= T() + T() + T() + T() + c().n + m.c.n

= T() + T() + T() + T() + c(α + β)m.n + m.c.n

= T() + T() + T() + T() + c.n + m.c.n

= T() + T() + T() + (m+1).c.n

= T() + T() + T() + (m+1).c.n

= T() + (m + 1).c.n

Vậy (\*) đúng với mọi m.

Công thức được thực hiện cho đến khi: = 1. Nó thực hiện tối đa số lần mà: βm.n = 1 ⬄⬄ m =

Và tối thiểu số lần: αm .n = 1 ⬄ = n ⬄ m = .

Khi mà: 1 thì:

T(n) = + m.c.n

= + m.c.n = + m.c.n

Ta có:

Suy ra: T(n) = .

1. **Cho dãy Fibonacci**

Chứng minh quy nạp theo n là: trong đó

Giải:

Với m = 0:

Với m = 1:

Giả sử ta chứng minh rằng

Thạt vậy áp dụng giả thiết quy nạp ta có:

Ta thấy

Do đó:

Mặt khác suy ra

Vậy ta kết luận

1. ***Bài 231-232 (trang 38) sách Ian Parberry - Problems on Algorithms.pdf***
2. **State anh prove a general formula for recurrences of the form**

****

Để dễ hình dung, ta chỉ xét các số n có dạng :



Ta có :



Tổng quát cho n bất kỳ thì đặt : , trong đó , và ; tức là : , làm tương tự như trên, ta được :



Vậy công thức tổng quát của dãy :



1. **State anh prove a general formula for recurrences of the form**

****

Giải

Để dễ hình dung, trước hết ta xét các số n có dạng :



Ta có :

****

Dãy ak-1c2, ak-2c4, ..., a2c2(k - 2), ac2(k - 1), c2k là cấp số nhân với công bội là . Do đó :

****

Trường hợp tổng quát, ta lấy , công thức tổng quát là:

****

***Bài 2:***

1. **Chuyển một số thập phân sang dạng nhị phân**

Ta chỉ quan tâm đến số phép toán thực hiện của chương trình đệ quy Int2Bin(n), kí hiệu là .Ta thấy:

Suy ra .

1. **Phân tích một số ra thừa số nguyên tố**

Trước khi vào chương trình phân tích nguyên tố, chương trình thực hiện các lệnh 21, 22, 23, 24

Giả sử số trong đó

Vào ptnt(n) vòng lặp while thực hiện bước lặp để tăng giá trị i từ 2 lên do nên ra khỏi while.

Sau đó, lệnh 13, 14 được thực hiện lần

Ta lại vào ptntvòng lặp while được thực hiện lần để tăng i từ lên

Sau đó lệnh 13, 14 được thực hiện lần

Ta lại vào

Quá trình trên được lặp lại k lần

Số lần thực hiện trong chương trình là

Ta có thể đánh giá

Còn , trong trường hợp xấu nhất nếu

Nói chung là , với

Ta đánh giá độ phức tạp của thuật toán theo trường hợp xấu nhất

1. **Số Fibonacci thứ n**

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

Ta có: với

Đặt

dãy U(n) có dạng tổng quát

trong đó là nghiệm của phương trình

Ta thấy nên

trong đó

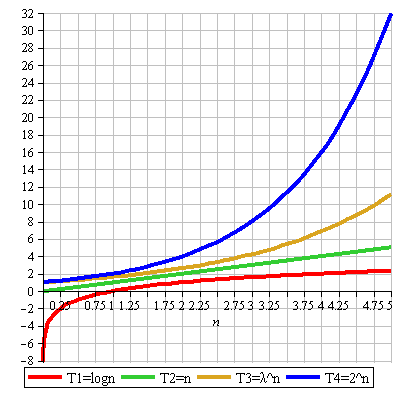
1. **Tháp Hà Nội**

Gọi là số lệnh thực hiện trong HanoiTower(n)

Ta có công thức truy hồi:

với

1. **Biểu đồ các hàm đánh giá**

****

***Bài 3:***

1. **Tìm kiếm nhị phân trên mảng được sắp**

Ta có trong trường hợp suy biên .Việc tìm kiếm một số x cho trước quy về việc so sánh x với phần tử duy nhất trong mảng.Nên với .Trong trường hợp bài toán quy về việc tìm phần tử x trên 2 mảng con được chia đôi từ mảng ban đầu cộng với việc so sánh phần tử x với phần tử chốt(phần tử ở vị trí giữa của mảng) nên .Tóm lại ta có thời gian thực hiện tìm kiếm nhị phân trên mảng đã sắp như sau:

Giải công thức trên ta được

1. **Tìm Max,Min của dãy số**

Ta có trong trường hợp suy biến dãy đã cho chỉ có 2 phần tử.Do vậy, ta chỉ cần tìm max, min giữa 2 phần tử đó.Do vậy với .Trong trường hợp , ta có bài toán chuyển thành tìm max,min ở 2 dãy con được chia bởi dãy ban đầu.Do vậy, .

Vậy ta có thời gian thực hiện thuật toán tìm max,min trên một dãy số như sau:

Giải công thức này ra ta có:

1. **Tìm cặp điểm gần nhau nhất trên mặt phẳng**

Thuật toán của ta gồm 2 bước:

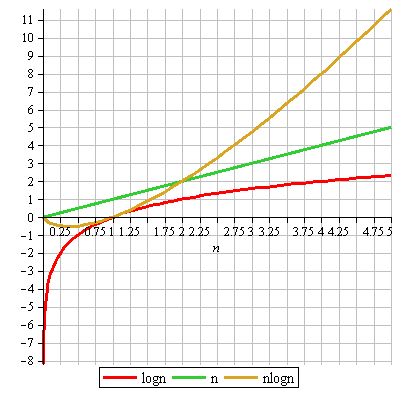
B1: Sắp xếp các điểm theo thứ tự tăng dần theo tọa độ x, ở đây ta dùng thuật toán QuickSort. Độ phức tạp của thuật toán là

B2: Sau khi sắp xếp xong thứ tự các điểm, ta tiến hành tìm cặp điểm gần nhất theo phương pháp chia để trị. Ta đánh giá theo trường hợp trung bình, khi đó tập n điểm bị chia thành 2 tập với lực lượng mỗi tập = n/2 (chốt giữa là ). Trong trường hợp trung bình, thì số điểm rơi vào khoảng là không đáng kể. Do đó ta có thể đánh giá

Đánh giá này cho ta

Tổng hợp 2 bước lại thì độ phức tạp thuật toán là:

1. **Biểu đồ các hàm đánh giá**



***Bài 4:* Chủ đề:”*Bài toán chia phần thưởng*”.**

* **Phân tích bài toán:**

Gọi là số cách chia m phần thưởng cho n học sinh.Trước hết, ta xét các trường hợp đơn giản.

* :Chỉ có 1 cách chia là tất cả học sinh đều nhận được 0 phần thưởng suy ra
* :Không có học sinh nào ta quy ước luôn

Ta sẽ tìm một công thức truy hồi cho để thiết kế thuật giải đệ quy cho bài toán này.

Xét bạn học sinh a có thành tích học tập kém nhất.Khia chia phần thưởng sẽ có 2 khả năng xảy ra:

* Bạn A không nhận được phần thưởng nào, khi đó phần thưởng sẽ được chia bạn còn lại.Như vậy, số cách chia phần thưởng mà trong đó bạn A không nhận được phần thưởng nào là:
* Bạn A nhận được ít nhất một phần thưởng .Do đó, A có thành tích kém nhất nên tất cả mọi học sinh đều nhận được ít nhất một phần thưởng.Nếu ta lấy đi của mỗi bạn 1 phần thưởng thì còn lại phần thưởng và sẽ không còn ràng buộc là bạn A phải có ít nhất một phần thưởng.Như vậy, số cách chia m phần thưởng cho n học sinh mà trong đó A nhân được không ít hơn một phần thưởng là:

Vậy ta có công thức truy hồi:

Khi sử dụng công thức truy hồi (\*) liên tiếp nhiều lần.Có thể đẫn đến trường hợp phải đánh giá với .Trong trường hợp này thì người có thành tích kém nhất sẽ nhận được 0 phần thưởng.(Vì trái lại, nếu người này nhận được hơn hay bằng một phần thưởng suy ra mỗi người có thành tích tốt hơn sẽ nhận được ít nhất một phần thưởng.Suy ra, tổng số phần thưởng ban đầu phải (Mâu thuẫn))

Do đó, trường hợp này chỉ cần phải xét xem chia phần thưởng cho người có thành tích tốt nhất mà thôi.Suy ra:

* **Xây dựng giải thuật:**

|  |  |
| --- | --- |
| 1 |  |
| 2 | If |
| 3 | If |
| 4 | If |
| 5 |  |
| 6 |  |
| 7 | End. |
| 8 | Main() |
| 9 | Input(m,n); |
| 10 | Result= |
| 11 | Output(Result) |
| 12 | End. |