**IV.** Tự đặt một số đề bài toán, phân tích bài toán, xây dựng giải thuật, đánh giá độ phức tạp và viết chương trình để minh họa phương pháp thiết kế giải thuật đệ quy và phương pháp chia để trị (ít nhất là 1 bài)

**Bài toán chia phần thưởng :**

***Đề Bài :***

Có một vật thưởng được chia cho n học sinh giỏi có xếp hàng theo thứ tự từ 1 đến n . Hỏi có bao nhiêu cách chia các phần thưởng thỏa mãn các điều kiện sau :

1. Học sinh giỏi hơn có số thưởng không ít hơn bạn kém hơn .
2. M vật thưởng phải chia hết cho các học sinh .

***Phân tích đề bài :***

In put : Hai số m và n (m >= n)

Out put : Các cách chia.

***Ý tưởng bài toán :***

Bài toán là việc phân tích số m thành tổng của n số . Sao cho số sau không lơn hơn số trước .

***Phân tích bài toán :***

Bài toán áp dụng phương pháp chia để trị

Có i phần thưởng mà có 0 học sinh thì có 0 cách chia.

Có 0 phần thưởng mà chia cho i học sinh thì có 1 cách chia (không ai có phần thưởng cả )

Nếu phần thưởng j mà ít hơn học sinh i thì : từ người thứ j + 1 không có phần thưởng vậy khi đó số cách chia sẽ là số cách chia j phần thưởng cho j học sinh .

Nếu số phần thưởng j và nhiều hơn số học sinh i thì có hai TH :

TH1 : Người cuối cùng không có phần thưởng, tức là chỉ chia j phần thưởng cho i - 1 người.

TH2: Người cuối cùng chắc chắn có phần thưởng khi đó ta sẽ lấy i phần thưởng chia cho i người, mỗi người sẽ có một phần thưởng trước , lúc này ta còn j – i phần thưởng , tiếp tục lấy số còn lại chia cho i người .

***Tính dừng :***

Bài toán chia phần thưởng của từng nhánh dừng khi có số học sinh hoặc số phần thưởng bằng 0.

Với số học sinh bằng 0 thì ta không có cách chia nào thỏa mãn, còn nếu số phần thưởng bằng 0 thì có một cách chia là không học sinh nào được phần thưởng(số phần thưởng mọi học sinh bằng 0)

***Đánh giá thời gian độ phức tạp :***

int chiaPhanThuong(int m,int n)

{

if(m == 0) return 1;

if( n==0) return 0;

if( m < n) return chiaPhanThuong(m,m);

else return chiaPhanThuong(m-n,n) + chiaPhanThuong(m,n-1);

}

Vậy độ phức tạp của thuật toán được thực tính theo công thức truy hồi :

Gọi T[i][j] là thời gian thực hiện chia i phần thưởng cho j học sinh khi đó ta có thể viết công thức truy hồi :

T[i][j] = T[i][j] + T[i][j] với (i>=j)

= T[i][i] (i<j)

= 1 nếu i = 0 hoặc j = 0

**Bài toán xếp ba lô.**

***Đề bài :***

Có n đồ vật , mỗi vật có trọng lượng Pi, và giá trị Vi (i = 1…….n)

Có một chiếc ba lô có thể chứa trọng lượng tối đa là M .Hãy xác định từng giá trị lớn nhất của các vật có thể đưa vào ba lô . Chỉ ra 1 các cách cho vật vào ba lô với tổng giá trị là lớn nhất.

***Ý tưởng bài toán:***

Chúng ta dùng thuật toán vét cạn

Giả sử ta chọn đồ vật thứ i, trong giới hạn trọng lượng , cập nhật tổng trọng lượng và giá trị .

Lặp lại quá trình chọn cho tới khi không thêm được nữa , kiểm tra lại cách này xem có tốt hơn không ?

Sau đó bỏ chọn từng đồ vật 1 , mỗi bước bỏ lại rẽ nhánh theo một hướng chọn cách khác nhau

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

Thực chất đây là bài toán tìm max. Danh sách các đồ vật được xếp theo thứ tự giảm dần theo đơn giá để dễ phân nhánh.

***Đánh giá thời gian độ phức tạp :***

Vòng lặp sẽ thực hiện hết các trường hợp có thể .

Nên thời gian thực hiện là số lần lặp sẽ là O(n!)

**Bài toán phân công công việc :**

Có n người với n công việc , người thứ i làm việc thứ j với năng suất a[i,j] .Hãy tìm cách phân công xem người nào làm việc nào thì tổng năng suất là tốt nhất .

***Phân tích đề bài :***

Input : n, ma trận năng suất

Outout : một cách phân công .

***Phân tích thuật giải :***

Duyệt tất cả các cách phân công và chọn ra cách tốt nhất .

Vẫn dùng thuật toán quay lui.

***Phân tích ý tưởng :***

Xét lần lượt từng công nhân , mỗi công nhân sẽ thử một công việc , nếu đã phân chia hết thì xem tổng năng suất có hơn cách đang có không ?

Lúc nào ta cũng có một công nhân chọn một công việc cố định trước (ta đánh dấu công việc đã có công nhân chọn) sau đó sẽ chọn lần lượt các công nhân tiếp theo với từng công việc khi đã duyệt xong ta sẽ chọn phương án tốt nhất.

***Đánh giá thời gian độ phức tạp :***

Người thứ nhất có n ngưởi cách chọn công việc thực hiện

Người thứ 2 có n -1 cách chọn công việc .

Người thứ i thì có (n-i) cách chọn …….

Đến người thứ n còn 1 cách chọn.

Vậy số lần gọi select là :

n(n-1)(n-2)….1 = n!

Vậy thời gian thực hiện bài toán là O(n!)

Biểu đồ biểu diễn thời gian thực hiện thuật toán O(n).

