**Chủ đề 1: Bài toán Knapsack**

**BÀI TOÁN 1: Cho n đồ vật. Đồ vật thứ i có khối lượng a [ i ] , giá trị b [ i ] . Một cái túi có sức chứa tối đa là M. Chọn 1 số đồ vật để cho tổng giá trị lấy được là lớn nhất.**

**\*\*\*\*Cách 1: duyệt n <= 20**

* **Mô tả nghiệm dưới dạng 1 dãy : x [ 1……..n ] ;**
* **x [ i ] = 1: chọn**
* **x [ i ] = 0: không chọn**
* **a [ 1 ] \* x [ 1 ] + …… + a [ n ] \* x [ n ] <= M ///a[i] là khối lượng đồ**
* **b [ 1 ] \* x [ 1] + ……. + b [ n ] \* x [ n ] -> max/// b[i] là giá trị đồ**
* **x [ i ] <- { 0 , 1 }**

**\*\*\*\*Cách 2: quy hoạch động :::: phát triển phương pháp quy nạp**

**F [ i ] [ j ] là giá trị lớn nhất có thể khi xét i đồ vật đầu tiên và kích cỡ túi bằng j.**

**F [ i ] [ j ] là giá trị của bài toán:**

**b [ 1 ] \* x [ 1 ] +……+ b [ i ] \* x [ i ] -> max**

**a [ 1 ] \* x [ 1 ] +……+ a [ i ] \* x [ i ] <= j**

**x [ i ] <- { 0 , 1 }**

**Công thức truy hồi: chỉ quan tâm đến x [ i ]**

**TH1 : x [ i ] = 0 : không chọn món đồ này**

* **a [ 1 ] \* x [ 1 ] + …… + a [ i - 1 ] \* x [ i - 1 ] <= j**
* **b [ 1 ] \* x [ 1 ] +…….+ b [ i - 1 ] \* x [ i - 1 ] -> max**

**🡺 kq = F [ i - 1 ] [ j ]**

**TH2 : x [ i ] = 1 : chọn món đồ này**

* **a [ 1 ] \* x [ 1 ] + …… + a [ i - 1 ] \* x [ i - 1 ] <= j - a [ i ]**
* **b [ 1 ] \* x [ 1 ] + …... + b [ i -1 ] \* x [ i - 1 ] + b [ i ] -> max**

**🡺 kq = F [ i - 1 ] [ j - a [ i ] ] + b [ i ]**

**🡺công thức:**

**F [ i ] [ j ] = max ( F [ i - 1 ] [ j ] , F [ i - 1 ] [ j - a [ i ] ] + b [ i ] )**

**Neo: F [ 0 ] [ j ] = 0 với mọi j**

**Đáp số : F [ n ] [ M ]**

**Code**

**// xây dựng bảng phương án**

***int F [ maxM ] ;***

***F [ 0 ] = 0;***

***Forinc ( i , 1 , n )***

***Fordec ( j , M , a[i] )***

***F [ j ] = max ( F [ j ] , F [ j - a [ i ] ] + b [ i ] ) ;***

***// truy vết***

***v = M ;***

***Fordec ( u , n , 1 )***

***{***

***x [ u ] = 0;***

***if ( v >= a [ u ] && F [ v ] == F [ v - a [ u ] ] + b [ v ] )***

***{***

***x [ u ] = 1 ;***

***v -= a [ u ] ;***

***}***

***}***

***Forinc ( i , 1 , n ) if ( x [ i ] == 1 ) cout << i << “ “ ;***

**Bài toán #2: như trên nhưng cho b [ i ] = a [ i ]**

* **a [ 1 ] \* x [ 1 ] + ……+ a [ n ] \* x [ n ] <= M**
* **a [ 1 ] \* x [ 1 ] + ……+ a [ n ] \* x [ n ] -> max**

**🡺tìm giá trị max có thể với điều kiện <= m**

**Bài toán chuyển thành: cho n số tự nhiên a [ 1.. n ] , tìm số lớn nhất bé hơn m có thể biểu diễn dưới dạng tổng các số trên.**

**F [ i ] [ j ] là ....................**

**Công thức truy hồi: F [ i ] [ j ] = max ( F [ i - 1 ] [ j ] , F [ i - 1 ] [ j - a [ i ] ] + a [ i ] )**

**Neo: F [ 0 ] [ j ] = 0 với mọi j**

**i : 1 -> n**

**j : a [ i ] -> m**

**BÀI TOÁN #3:**

* **a [ 1 ] \* x [ 1 ] + …… + a [ n ] \* x [ n ] <= M**
* **b [ 1 ] \* x [ 1 ] + …… + b [ n ] \* x [ n ] -> max**
* **x [ i ] <- N : N loại đồ vật**

**F[i][j] là giá trị hàm mục tiêu**

**TH1 : x [ i ] = 0 --> F [ i ] [ j ] = F [ i - 1 ] [ j ]**

**TH2: x [ i ] ! = 0**

* **a[ 1 ] \* x [ 1 ] + …… + a [ i - 1 ] \* x [ i - 1 ] + a [ i ] \* ( x [ i ] - 1 ) <= j - a [ i ]**
* **b[ 1 ] \* x [ 1 ] + …… + b [ i - 1 ] \* x [ i - 1 ] + b [ i ] \* ( x [ i ] - 1 ) + b [ i ] -> max**

**F [ i ] [ j ] = F [ i ] [ j - a [ i ] ] + b [ i ]**

**🡺công thức truy hồi : F [ i ] [ j ] = max ( F [ i - 1 ] [ j ] , F [ i ] [ j - a [ i ] ] + b [ i ] )**

**Neo F [ 0 ] [ j ] = 0 với mọi j**

**Code**

**// xây dựng bảng phương án**

***Forinc ( i , 1 , n )***

***Forinc ( j , a [ i ] , M ) F [ i ] [ j ] = max ( F [ i - 1 ] [ j ] , F [ i ] [ j - a [ i ] ] + b [ i ] )***

**// tìm phương án**

***j = M;***

***Forinc ( i , 1 , n ) x [ i ] = 0 ;***

***Fordec ( i , n , 1 )***

***{***

***While ( j >= a [ i ] && F [ i ] [ j ] == F [ i ] [ j - a [ i ] ] + b [ i ] )***

***{***

***x [ i ] ++ ;***

***j - = a [ i ] ;***

***}***

***}***

**BÀI TOÁN #4:**

* **a [ 1 ] \* x [ 1 ] + …… + a [ n ] \* x [ n ] = M**
* **b [ 1 ] \* x [ 1 ] + …… + b [ n ] \* x [ n ] -> max**
* **x [ i ] <- { 0 , 1 }**

**F [ i ] [ j ] là giá trị hàm mục tiêu ; gán F [ i ] [ j ] = -oo nếu không có nghiệm**

**F [ i ] [ j ] = max ( F [ i - 1 ] [ j ] , F [ i - 1 ] [ j - a[ i ] ] + b[ i ] ( với ( F [ i - 1 ] [ j - a[ i ] ] != -oo ) )**

**neo: F [ 0 ] [ j ] = 0 nếu j = 0**

**= -oo nếu j > 0**

**NẾU: a [ i ] = b [ i ]**

* **a [ 1 ] \* x [ 1 ] + …… + a [ n ] \* x [ n ] = M**
* **a[ 1 ] \* x [ 1 ] + …… + a [ n ] \* x [ n ] -> max**
* **x [ i ] <- { 0 , 1 }**
* **Bài toán đánh dấu tổng**

**Code**

**// xây dựng bảng phương án**

***F [ 0 ] = 0 ;***

***Forinc ( j , 1 , m ) F [ j ] = -oo ;***

***Forinc ( i , 1 , n )***

***Fordec ( j , M , a [ i ] )***

***If ( F [ j - a [ i ] ] != -oo ) F [ j ] = i ;***

**// tìm ngược**

***j = M;***

***while ( j > 0 )***

***{***

***x [ F [ j ] ] = 1 ;***

***j -= a [ F [ j ] ] ;***

***}***

**BÀI TOÁN #5:**

* **a [ 1 ] \* x [ 1 ] + ……+ a [ n ] \* x [ n ] <= M**
* **b [ 1 ] \* x [ 1 ] + ……+ b [ n ] \* x [ n ] -> max**
* **x [ i ] <- { 0 , 1 , ....... , k }**

**F [ i ] [ j ] là giá trị hàm mục tiêu**

* **a [ 1 ] \* x [ 1 ] +…… + a [ i ] \*x [ i ] <= j**

**b [ 1 ] \* x [ 1 ] +…… + b [ i ] \* x [ i ] -> max**

**k + 1 trường hợp x [ i ] = u (u = 0 . . . . k )**

**công thức truy hồi: F [ i ] [ j ] = max ( F [ i - 1 ] [ j - u \* a [ i ] ] + u \* b [ i ] ) ( 0 < = u < = k )**

**neo : F [ 0 ] [ j ] = 0**

**BÁI TOÁN #6: đếm xem có bao nhiêu cách chọn các số để có tổng là M**

* **a [ 1 ] \* x [ 1 ] + …… + a [ n ] \* x [ n ] = M**

**F [ i ] [ j ] là số cách để đánh dấu được tổng j mà sử dụng a [ 1.....i ]**

**TH1: không chọn a [ i ] F [ i ] [ j ] = F [ i - 1 ] [ j ]**

**TH2: chọn a [ i ] F [ i ] [ j ] = F [ i - 1 ] [ j - a [ i ] ]**

**🡺F [ i ] [ j ] = F [ i - 1 ] [ j ] + F [ i - 1 ] [ j - a [ i ] ]**

**Mở rộng: mỗi số chọn <= k lần**

**F [ i ] [ j ] += F [ i - 1 ] [ j - u \* a [ i ] ] u : 0 .......... k**

**\*chú ý: F [ 0 ] [ 0 ] = 1**