**MỤC LỤC**

[I. Nêu bài toán 2](#_Toc501144135)

[II. Mô tả chi tiết thuật toán 2](#_Toc501144136)

[III. Đánh giá độ phức tạp thuật toán 4](#_Toc501144137)

[IV. Thực hiện từng bước của thuật toán trên 2 bộ dữ liệu với (N>=5) 4](#_Toc501144138)

# **I. Nêu bài toán**

Bài toán cái túi (hay còn gọi là bài toán xếp ba lô) là một bài toán tối ưu tổ hợp. Bài toán được đặt tên từ vấn đề chọn những gì quan trong có thể nhét vừa vào một cái túi (với giới hạn khối lượng) để mang theo trong một chuyến đi.

Ví dụ về bài toán cái túi trong giới hạn một chiều: chọn các hộp nào để làm cho giá trị tiền là cực đại khi tổng trọng lượng không vượt quá 15kg? Bài toán đa chiều có thể xét đến khối lượng riêng và kích thước của hộp, đó là bài toán xếp vali điển hình (packing problem). Ta có n loại đồ vật, từ 1 tới n. Mỗi đồ vật thứ i có trọng lượng w[i] và giá trị v[i]. Trọng lượng tối đa mà túi có thể mang được là S.

Bài toán cái túi dạng 0/1 thì hạn chế số đồ vật thuộc mỗi loại là 0 (không được chọn) và 1 (được chọn).

# **II. Mô tả chi tiết thuật toán**

\*Thuật toán tạo bảng

Trường hợp đơn giản có duy nhất 1 vật để chọn. Ta tính K [1, j] với mọi j. Nếu có thể chọn (nghĩa là khối lượng tối đã của balo >= khối lượng của vật thứ 1) thì chọn

K [1, j]=weight[1] . Ngược lại (j<weight[i]) không thể chọn, nghĩa là K [1, j] = 0. i là vật thứ 1,2,3,4,5 để ta chọn i<=n (n là số vật ta có ). J là số cân nặng ta xét

j<=w (w là số cân tối đa ta xếp vào balo)

Nếu i=0 hoặc j=0 thì K[i] [j] =0 giá trị của vật đó bằng 0

Nếu j< weight[i] nghĩa là số cân ta đang xét nhỏ hơn số cân của vật thứ i, thì K[i] [j] = K [i-1] [j] giá trị ô đang xét bằng giá trị ô bên trên nó tương ứng

Ngược lại j> weight[i], ta tính x= j- weight[i]. K[i-1][j] > worth[i]+ K[i-1][x] phần tử bên trên lớn hơn giá trị của vật thứ i + K[i-1][x] thì vẫn hạ giá trị phần tử bên trên xuống K[i][j]= K[i-1][j] . Nếu K [i-1] [j] < worth[i] + K[i-1][x] thì K[i][j] = K[i-1][x] + worth

\*Thuật toán tra bảng để tìm ra các vật được chọn

Chú ý: Nếu K [i, j] = K [i–1, j] thì món thứ i không được chọn. K [n, W] là tổng khối lượng tối ưu của các món hàng bỏ vào ba lô.

Bước 1: Bắt đầu từ i = n, j = W.

Bước 2: Tìm trong cột j, ngược từ dưới lên, ta tìm dòng i sao cho K [i,j] > K[i–1, j]. Đánh dấu món thứ i được chọn: Chọn[i] = true;

Bước 3: j = K [i, j] – weight[i]. Nếu j> 0 thì thực hiện bước 2, ngược lại thực hiện bước 4

Bước 4: Dựa vào mảng chọn để in ra các món hàng được chọn.

Ví dụ cho bảng giá trị

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cân | 1 | 3 | 4 | 5 |
| Gía trị | 1 | 4 | 5 | 7 |

Dưới đây là bảng tính được ở ví dụ trên

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| K[i,j] | J=0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| I=0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 0 | 1 | 1 | 4 | 5 | 5 | 5 | 5 |
| 3 | 0 | 1 | 1 | 4 | 5 | 6 | 6 | 9 |
| 4 | 0 | 1 | 1 | 4 | 5 | 7 | 8 | 9 |

# **III. Đánh giá độ phức tạp thuật toán**

Thuật toán sử dụng vòng for là n trong vòng for có 3 bước 2 bước so sánh if và 1 bước gán j. Nên độ phức tạp của thuật toán là O (n)

# **IV. Thực hiện từng bước của thuật toán trên 2 bộ dữ liệu với (N>=5)**

* Bộ dữ liệu thứ nhất

Cho bảng giá trị, xếp các vật vào balo sao cho tối đa balo chứa được 15kg

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Cân | 12 | 2 | 1 | 1 | 4 |
| Gía trị | 4 | 2 | 1 | 2 | 10 |

Bảng kết quả sau khi sử dụng thuật toán để tìm ra những vật được chọn và xếp vào balo. Kết quả là chọn các vật thứ 5,4,3,2 để xếp vào balo

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| K[i,j] | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 4 | 4 | 4 |
| 2 | 0 | 0 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 4 | 4 | 6 | 6 |
| 3 | 0 | 1 | 2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 4 | 5 | 6 | 7 |
| 4 | 0 | 2 | 3 | 4 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 6 | 7 | 8 |
| 5 | 0 | 2 | 3 | 4 | 10 | 12 | 13 | 14 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 |

Đầu tiên ta tạo bảng như trên để tìm ra các vật được chọn, i là vật thứ 1, 2, 3, 4, 5 để ta chọn i<=n (n là số vật ta có). j là số cân nặng ta xét j<=w (w là số cân tối đa ta xếp vào balo)

Nếu i=0 hoặc j=0 thì K[i] [j] =0 giá trị của vật đó bằng 0. Tất cả cột i=0, j=0 thì điền giá trị 0

Xét vật thứ nhất (i=1) có khối lượng 12kg , j từ 0->11 nhỏ hơn 12 nên K[i][j]= K[i-1][j] các phần tử điền hết giá trị là 0 . Đến j=12 thì thấy giá trị là 4, tương tự j=13, 14, 15 tương tự điền 4

Xét vật thứ hai (i=2) j=2 ta thấy có giá trị 2 nên điền 2 vào ô , j từ 3->11 lớn hơn weight(2) nên tính x=j-weight(2), do K[1][j]<worth[2]+ K[1][x] thì K[2][j]= K[1][x]+worth[2]=2 (j từ 2->11) . j=14, 15 tính tương tự ta điền được 6. Cứ tương tự ta điền được hết các giá trị vào bảng và tìm được những vật được chọn xếp vào balo

* Bộ dữ liệu thứ 2

Cho bảng giá trị, xếp các vật vào balo sao cho tối đa balo chứa được 12kg

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Cân | 2 | 1 | 4 | 1 | 5 |
| Gía trị | 3 | 5 | 2 | 7 | 1 |

Bảng kết quả sau khi sử dụng thuật toán để tìm ra những vật được chọn và xếp vào balo. Kết quả các vật chọn là vật thứ 4,3,2,1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| K[i,j] | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |  |  |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |
| 1 | 0 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |  |  |
| 2 | 0 | 5 | 5 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |  |  |
| 3 | 0 | 5 | 5 | 8 | 8 | 8 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |  |  |
| 4 | 0 | 7 | 12 | 12 | 15 | 15 | 15 | 15 | 17 | 17 | 17 | 17 | 17 |  |  |
| 5 | 0 | 7 | 12 | 12 | 15 | 15 | 15 | 15 | 17 | 17 | 17 | 17 | 17 |  |  |