**Unbounded Knapsack**

1. **Bài toán**

Một kẻ trộm đột nhập vào một cửa hiệu tìm thấy có n mặt hàng có trọng lượng và giá trị khác nhau, nhưng hắn chỉ mang theo một cái túi có sức chứa về trọng lượng tối đa là M. Vậy kẻ trộm nên bỏ vào ba lô những món nào và số lượng bao nhiêu để đạt giá trị cao nhất trong khả năng mà hắn có thể mang đi được.

Câu hỏi là: Có thể đạt được một giá trị nhiều nhất là bao nhiêu

Ta có n loại đồ vật, x1 tới xn. Mỗi đồ vật xi có một giá trị vi và một khối lượng wi. Khối lượng tối đa mà ta có thể mang trong balo là C.

Bài toán sắp xếp balo dạng 0-1: Hạn chế số đồ vật thuộc mỗi loại là 0 hoặc 1.

Bài toán sắp xếp balo bị chặn: Hạn chế số đồ vật thuộc mỗi loại không vượt quá một số lượng nào đó.

Bài toán sắp xếp balo không bị chặn: không có một hạn chế nào về số lượng đồ vật.

Bài toán sắp xếp balo dạng phân số: Với mỗi loại có thể chọn một phần của nó (ví dụ tên trộm đánh cắp 4kg thịt bò có thể cắt thành nhiều phần để bỏ vào balo)

Bài toán sắp xếp balo thường được giải bằng quy hoạch động.

Input:

* Dòng thứ nhất là hai số N, C (1 ≤ N ≤ 20, 1 ≤ M ≤ 100)
* Trong N dòng kế tiếp, dòng thứ I là 2 số nguyên wi và vi (1 ≤ wi ≤ C, 1 ≤ vi ≤ 100)

Output

* Nếu tên trộm không thể lấy được món đồ nào, in ra 0.
* Nếu tên trộm có thể lấy được ít nhất một món đồ, dòng thứ nhất in ra giá trị lớn nhất tên trộm có thể lấy, dòng thứ hai là chỉ số những thứ mà tên trộm đã lấy. Nếu có nhiều cách lấy đồ có cùng giá trị lớn nhất, chỉ cần in ra một cách lấy đồ bất kỳ.

1. **Mô tả chi tiết thuật toán**

Thuật toán Unbounded Knapsack (Trường hợp không giới hạn số lượng mỗi vật)

Gọi các chi phí là w1,…, wnvà các giá trị tương ứng là v1,..vn. Ta cần cực đại hóa tổng giá trị với điều kiện tổng chi phí không vượt quá C. Khi đó, với mỗi i ≤ C, đặt A(i, j) là giá trị lớn nhất có thể đạt được với tổng khối lượng không vượt quá j.

Nếu không chọn vật thứ i thì A(i, j) là giá trị lớn nhất có thể chọn trong số các vật.

A(i, j) = A(i – 1, j) (1)

Nếu chọn vật thứ j (Phải thỏa mãn wj ≤ i) thì ta có A(i, j) bằng giá trị vật thứ j là vj cộng với giá trị lớn nhất có thể trọng trong số các vật {1, 2,..., j – 1}

A(i, j) = vj + A(i, j - wj) (2)

Vấn đề đặt ra là phải xem xét (1) hay (2) là phương án tốt hơn, từ đó ta có công thức truy hồi:

A(0) = 0

A(i) = max { A(i, j) = vj + A(i, j – wj), A(i – 1, j)}

**Truy vết:**

Bắt đầu từ A( n, M ), nếu f[ n, M ] = f[ n-1, M] thì tức là không chọn vật thứ n, ta truy về f[ n – 1, M ]. Còn nếu f[ n, M ] ≠ f[ n-1, M ] thì ta thông báo rằng phép chọn tối ưu có chọn vật thứ n và truy về f[ n, M-Wn]

1. **Độ phức tạp của thuật toán**

Ở đây, giá trị lớn nhất của tập rỗng được lấy bằng 0. Tính dần các kết quả từ A(0) tới A(C), ta sẽ được lời giải. Do việc tính mỗi A(i) đòi hỏi xem xét n đồ vật (tất cả các giá trị này đã được tính từ trước), và có C giá trị của các A(i) cần tính, nên thời gian chạy của lời giải quy hoạch động là O(nC).

1. **Ví dụ**

Ví dụ 1:

N = 5, M = 12

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| i | 1 | 2 | 3 | 4 | 5 |
| W(i) | 2 | 3 | 1 | 4 | 5 |
| V(i) | 5 | 7 | 2 | 10 | 15 |

Bảng phương án:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 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 | 0 | 5 | 5 | 10 | 10 | 15 | 15 | 20 | 20 | 25 | 25 | 30 |
| 2 | 0 | 0 | 5 | 7 | 10 | 12 | 15 | 17 | 20 | 22 | 25 | 27 | 30 |
| 3 | 0 | 2 | 5 | 7 | 10 | 12 | 15 | 17 | 20 | 22 | 25 | 27 | 30 |
| 4 | 0 | 2 | 5 | 7 | 10 | 12 | 15 | 17 | 20 | 22 | 25 | 27 | 30 |
| 5 | 0 | 2 | 5 | 7 | 10 | 15 | 17 | 20 | 22 | 25 | 30 | 32 | 35 |

**Truy vết:**

Bắt đầu từ ô A[ 5, 12 ]: A[ 5, 12 ] ≠ A[ 4, 13 ] => chọn vật thứ 5

=> xét ô A[ 5, 7 ] ≠ A[ 4, 7 ] =>chọn vật thứ 5

=> xét ô A[ 5, 2 ] = A[ 4, 2 ] => không chọn vật thứ 5

=> xét ô A[ 4, 2 ] = A[ 3, 2 ] => không chọn vật thứ 4

=> xét ô A[ 3, 2 ] = A[ 2, 2 ] => không chọn vật thứ 3

=> xét ô A[ 2, 2 ] = A[ 1, 2 ] => không chọn vật thứ 2

=> xét ô A[ 1, 2 ] = A[ 0, 2 ] => chọn vật thứ 1

Vậy ta chọn 2 vật số 5 và 1 vật số 1 để có giá trị max là 35 thỏa mãn khối lượng nhỏ hơn hoặc bằng M = 12