Bài toán cái túi theo phương pháp nhánh cận

**I. Bài toán**

Có n đồ vật, mỗi loại có số lượng không hạn chế. Đồ vật loại I, đặc trưng bởi trọng lượng Wi và giá trị sử dụng Vi, với mọi I € { 1,…,n }.

Cần chọn các vật này đặt vào một chiếc túi xách có giới hạn trọng lượng m, sao cho tổng giá trị sử dụng các vật được chọn là lớn nhất.

**II. Mô tả chi tiết thuật toán**

1. Ý tưởng

- Đặt : D ={u=()  : }

Và:

- Bài toán chiếc túi xách chuyển về dạng sau:

Tìm :

- Cho nên ta sẽ kết hợp đánh giá nhánh cận trong quá trình liệt kê cac lời giải theo phương pháp quay lui.

2. Thiết kế thuật toán

Try(i)

For(j=1t)

If(chấp nhận được)

{

Xác định  theo j;

Ghi nhận trạng thái mới;

If(i==n)

Cập nhật lời giải tối ưu;

Else

{

Xác định cận trên g;

If(g() )

Try(i+1);

}

Trả lại trạng thái cũ cho bài toán;

}

* Cách chọn vật:

Xét mảng đơn giá:

Ta chọn vật theo đơn giá giảm dần.

Không mất tính tổng quát, ta giả sử các loại vật cho theo thứ tự giảm dần của đơn giá.

* Đánh giá cận trên:

Giả sử đã tìm được lời giải bộ phận: () .Khi đó:

Cận trên cho các lời giải bộ phận cấp i có thể xác định bởi:

g() =

* Theo biểu thức xác định cận trên g, các giá trị có thể chấp nhận được cho là:

t=0

* Thao tác ghi nhận trạng thái mới khi xác định được chẳng qua là cập nhật lại giá trị thu được và giới hạn trọng lượng mới của chiếc túi:

S = S +

T = T +

* Vì vậy thao tác trả lại trạng thái cũ cho bài toán:

S = S -

T = T -

* Cập nhật lời giải tối ưu:

Khi tìm được một lời giải, ta so sánh lời giải này với lời giải má ta coi là tốt nhất vào thời điểm hiện tại để chọn lời giải tối ưu.

**III. Thực hiện thuật toán**

Bộ dữ liệu 1

|  |  |  |
| --- | --- | --- |
| i | w | v |
| 1 | 7 | 9 |
| 2 | 4 | 7 |
| 3 | 1 | 2 |
| 4 | 7 | 1 |
| 5 | 5 | 4 |
| 6 | 3 | 9 |
| 7 | 9 | 7 |
| 8 | 2 | 1 |
| 9 | 4 | 6 |
| 10 | 6 | 3 |

m = 20

Sắp xếp theo thứ tự giảm dần của đơn giá wv[i] =

|  |  |  |  |
| --- | --- | --- | --- |
| i | w | v | tyso |
| 1 | 3 | 9 | 3 |
| 2 | 1 | 2 | 2 |
| 3 | 4 | 7 | 1.75 |
| 4 | 4 | 6 | 1.5 |
| 5 | 7 | 9 | 1.29 |
| 6 | 5 | 4 | 0.8 |
| 7 | 9 | 7 | 0.78 |
| 8 | 2 | 1 | 0.5 |
| 9 | 6 | 3 | 0.5 |
| 10 | 7 | 1 | 0.14 |

Gốc

(5)

S=45

TL=15

g=51

(6)

S= 54

TL =18

g =58

(6,2)

S=58

TL=20

g=58

Cắt nhánh này vì:

g<

(6,1)

S=56

TL=19

g=57.75

(6,2,0)

S=58

TL=20

g=58

Lời giải tối ưu:

=(6,2,0,0,0,0,0,0,0,0)

(6,2,0,0)

S=58

TL=20

g=58

(6,2,0,0,0,0,0,0,0,0)

S=58

TL=20

**IV. Chương trình sử dụng C++**

//thuat toan nhanh can, tim phuong an toi uu

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <conio.h>

const int max = 100;

int w[max], v[max], n; //khai bao mang chua khoi luong, gia tri cua cac do vat, va n

float wv[max];

int soluong;//so luong

int khoiluong;//khoi luong túi

int khoiluongdovat;//khoi luong do vat chiem trong túi sau khi xep toi uu

int TL=0;//trong luong tong các vat da xep(hien tai)

int dem = 0;

int gt[100]; //cac buoc cua thuat toan

int sl[100],slmax[100]; //nhanh can

int s=0;//tong giá tri tai thoi diem hien tai

int fmax=0;//giá tri lon nhat sau moi lan duyet het

float g;//dieu kien bo nhanh

void input()

{

FILE \*f;

int i, x, y;

f = fopen("CaiTui1.txt", "r");

fscanf(f, "%d", &khoiluong);

fscanf(f, "%d", &n);

for (i = 0; i < n; i++)

{

fscanf(f, "%d", &x);

w[i] = x;

fscanf(f, "%d", &y);

v[i] = y;

wv[i] = (float)v[i] / w[i];

}

fclose(f);

}

void hoandoi(float &a, float &b) {

float temp = a;

a = b;

b = temp;

}

//sap xep theo thu tu giam dan cua v[i]/w[i] tuc la wv[i]

void sort() {

for (int i = 0; i<n; i++) {

for (int j = i + 1; j<n; j++) {

if (wv[j]>wv[i]) {

hoandoi((float&)w[j], (float&)w[i]);

hoandoi((float&)v[j], (float&)v[i]);

hoandoi(wv[j], wv[i]);

}

else if (wv[j] == wv[i])

{

if (w[j]<w[i])//sx tang dan cua khoi luong

{

hoandoi((float&)w[j], (float&)w[i]);

hoandoi((float&)v[j], (float&)v[i]);

hoandoi(wv[j], wv[i]);

}

}

}

}

}

//hien thi

void display() {

printf("\nDo vat khoiluong Giatri Tyso\n");

for (int i = 0; i<n; i++) {

printf("%d \t %3d \t %3d \t%5.2f", i+1, w[i], v[i], wv[i]);

printf("\n");

}

}

//thuat toan nhanh can

void Try(int i)

{

int j;

soluong = (khoiluong - TL) / w[i];

for (j = soluong; j >= 0; j--)

{

sl[i] = j;

TL += w[i] \* sl[i];

s += v[i] \* sl[i];

printf("%3d%3d%3d%3f\n",soluong,s,TL,g);

if (i == n-1)

{

if (s > fmax)

{

for (int k = 0; k < n; k++)

{

slmax[k] = sl[k];

}

khoiluongdovat = TL;

fmax = s;

}

}

else

{

g = s + (float)v[i + 1] \* (khoiluong - TL) / w[i + 1];

if (g > fmax)

Try(i + 1);

}

TL = TL - w[i] \* sl[i];

s = s - v[i] \* sl[i];

}

}

int main()

{

input();

//display();

printf("\nSau khi sap xep:\n");

sort();

display();

printf("\nThuat toan nhanh can: \n");

Try(0);

printf("\nDo vat soluong khoiluong giatri\n");

for (int i = 0; i<n; i++)

{

printf("%d %d %d %d \n", i+1,slmax[i],w[i],v[i]);

printf("\n");

}

printf("\nKhoi luong tui da dung: %d", khoiluongdovat);

printf("\nTong gia tri: %d", fmax);

getchar();

getchar();

return 1;

}

**V. Đánh giá độ phức tạp của thuật toán**