HỌC VIỆN KỸ THUẬT QUÂN SỰ

KHOA CÔNG NGHỆ THÔNG TIN



**BÀI TẬP MÔN HỌC**

PHÂN TÍCH VÀ THIẾT KẾ GIẢI THUẬT

**ĐỀ 22**

***SỬ DỤNG THUẬT TOÁN KNAPSACK ĐỂ LÀM BÀI TOÁN CÁI TÚI***

*Giáo viên hướng dẫn :****NGUYỄN MẬU UYÊN***

*Sinh viên thực hiện :* ***NGUYỄN THÙY LINH***

*Lớp :* ***CNTT14***

*Mã lớp học :* **12325151 1**

1. **Giới thiệu chung về bài toán cái túi**

* Bài toán cái túi là một bài toán tối ưu hóa. Bài toán được đặt tên từ vấn đề chọn những gì quan trọng để nhét vừa vào trong một cái túi (giới hạn về khối lượng) để mang theo trong một chuyến đi. Bài toán này cũng thương xuyên xuất hiện trong vấn đề kinh doanh, toán tổ hợp…
* Bài toán cái túi có 2 dạng:
* Dạng 0/1: Hạn chế về số lượng, được chọn là 1, không được chọn là 0.
* Dạng không bị giới hạn về số lượng đồ vật (unbounded knapsack)

1. **Mô tả chi tiết bài toán cái túi 0/1**
2. **Bài toán**

* Một kẻ trộm đột nhập vào một ngôi nhà tìm thấy các n món 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 L cân. Vì thế hắn phải lấy những món đồ nào sao cho giá trị là lớn nhất, tuy nhiên mỗi đồ chỉ được lấy một cái.

1. **Ý tưởng**

* Mô tả:

Input: W[k], V[k],n

Output: tìm được những vật xếp được vào balo sao cho có giá trị lớn nhất.

* Với n mặt hàng, S là khối lượng tối đa mà túi mang được, W[k] là trọng lượng đồ thứ k, V[k] là giá trị vật thứ k, L là tổng khối lượng các vật.
* Với mỗi 1 ≤ k ≤ n và 0 ≤ S ≤ L, xét bài toán Value(k,S). Tính Value[k,S] là tổng giá trị lớn nhất của các đồ vật được chọn trong số k đồ vật đầu tiên và có tổng trọng lượng không vượt quá S.
* Ta có

Value[k,S]=

* Khi đó để tính Value[k,S] ta có thể lập luận như sau:

+ Nếu S≤ thì vật thứ k không thể bỏ vào túi khi đó cách chất tối ưu là sử dụng k-1 đồ vật trước đó và có giá trị là Value[k-1,S]

**+** Nếu S ≥thì cần lựa chọn

Chất đồ thứ k vào túi, khi đó trọng lượng còn lại sẽ được chất vào túi từ k-1 đồ vật đầu tiên với giá trị là Value[k-1,S-].

* Từ đó ta có công thức:

Value[k,S]=

1. **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).

1. **Thực hiện các bước và kết quả mỗi bước**

* **Bộ thứ 1**
* Cho một cái túi chứa được tối đa 10 cân.
* Cho 5 đồ vật có giá trị như sau:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Cân | 6 | 2 | 4 | 3 | 11 |
| Giá trị | 20 | 8 | 14 | 13 | 35 |

Khi đó các bước tính lần lượt như sau:

* Bước 1: gọi i là vật, j là số cân của vật, v là giá trị vật

Khi không có vật nào thì sẽ không có cân nào được bỏ vào túi

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

* Bước 2: ta xét vật thứ nhất i1=1,

Xét vật 1 nặng 6 cân khi đó v(1,2,3,4,5) = 0 (không đủ để chứa vật 1)

v(6) = 20

* V(7,8,9,10) =20 do chỉ có một vật, mà vật 1 chỉ được lấy 1 lần.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 20 | 20 | 20 | 20 | 20 |

* Bước 3: i2=2,

Vật 2 nặng 2 cân => v(2,3,4,5)= 8

V(6,7) = 20

Tổng 2 vật nặng 8 cân => v(8,9,10)= 20+8=28

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 20 | 20 | 20 | 20 | 20 |
| 2 | 0 | 0 | 8 | 8 | 8 | 8 | 20 | 20 | 28 | 28 | 28 |

* Bước 4: i3 = 3

Vật 3 nặng 4 cân => v(2,3)= 8, v(4,5 )= 14

V(6,7) = v(2)+v (4)=22 > v (6)

Tổng 2 vật nặng 8 cân => v(8,9)= 20+8=28

v (10) = v (4) + v (6) = 34

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 20 | 20 | 20 | 20 | 20 |
| 2 | 0 | 0 | 8 | 8 | 8 | 8 | 20 | 20 | 28 | 28 | 28 |
| 3 | 0 | 0 | 8 | 8 | 14 | 14 | 22 | 22 | 28 | 28 | 34 |

* Bước 5:i4 =4

Vật 4 nặng 3 cân => v(2)= 8, v(3) = 13, v(4)= 14,

v(5) = v(2) + v(3)=21, v(6) = v(2)+v (4)=22 > v (6)

v(7) = v(3)+v (4)=27

Tổng 2 vật nặng 8 cân => v(8)= 20+8=28

v (9,10) = v (2) + v (3)+ v(4) = 35

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 20 | 20 | 20 | 20 | 20 |
| 2 | 0 | 0 | 8 | 8 | 8 | 8 | 20 | 20 | 28 | 28 | 28 |
| 3 | 0 | 0 | 8 | 8 | 14 | 14 | 22 | 22 | 28 | 28 | 34 |
| 4 | 0 | 0 | 8 | 13 | 14 | 21 | 22 | 27 | 28 | 35 | 35 |

* Bước 6: i5 = 5

Vật 4 nặng 3 cân => v(2)= 8, v(3) = 13, v(4)= 14,

v(5) = v(2) + v(3)=21, v(6) = v(2)+v (4)=22 > v (6)

v(7) = v(3)+v (4)=27

Tổng 2 vật nặng 8 cân => v(8)= 20+8=28

v (9,10) = v (2) + v (3)+ v(4) = 35

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 20 | 20 | 20 | 20 | 20 |
| 2 | 0 | 0 | 8 | 8 | 8 | 8 | 20 | 20 | 28 | 28 | 28 |
| 3 | 0 | 0 | 8 | 8 | 14 | 14 | 22 | 22 | 28 | 28 | 34 |
| 4 | 0 | 0 | 8 | 13 | 14 | 21 | 22 | 27 | 28 | 35 | 35 |
| 5 | 0 | 0 | 8 | 13 | 14 | 21 | 22 | 27 | 28 | 35 | 35 |

* Ta bắt đầu truy vết để tìm ra các vật được chọn

Nếu value[i][j] != value[i-1][j] thì ta lấy vật i

Cứ vậy lấy lần lượt các vật kia và số cân tối đa sẽ còn là s-w[i]

Thuật toán dừng khi value[i][j] = 0

* Áp dụng vào bài

Tại cân thứ 10 vật thứ 5 ta thấy value[5][10]=value[4][10] (35=35)

Không chọn vật thứ 5

* Trở về bài toán chọn 10 cân với số vật là 4

value[4][10] != value[3][10] => lấy vật thứ 4

* Khi đó số cân còn lại là s = 10-3=7 đưa về bài toán tối đa 7 cân với số vật là 3.

value[3][7] != value[2][7] => lấy vật thứ 3

* Khi đó số cân còn lại là s = 7-4=3 đưa về bài toán tối đa 3 cân với số vật là 2.

value[2][3] != value[1][3] => lấy vật thứ 2

* Khi đó số cân còn lại là s = 1 đưa về bài toán tối đa 1 cân với số vật là 1.

value[1][1] = value[2][7] => lấy vật thứ 3.

* Vậy do đó vật 4,3,2 được lấy
* **Bộ thứ 2**
* Cho một cái túi chứa được tối đa 8cân.

Cho 3 đồ vật có giá trị như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| Cân | 3 | 2 | 5 |
| Giá trị | 7 | 6 | 11 |

* Tương tự bài trên ta có luôn bảng quy hoạch động

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 7 | 7 | 7 | 7 | 7 | 7 |
| 2 | 0 | 0 | 6 | 7 | 7 | 13 | 13 | 13 | 13 |
| 3 | 0 | 0 | 6 | 7 | 7 | 13 | 13 | 17 | 18 |

* Lập luận tương tự ta chọn được vật 3,1.

1. **Chương trình**

#include <stdio.h>

#include <conio.h>

#include <iostream>

using namespace std;

void xuat(int Value[50][100], int n, int L)

{

cout<< endl;

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

{

for(int j = 0; j<= L; j++)

{

cout<< Value[i][j]<<" ";

}

cout<< endl;

}

}

void createtable(int Value[50][100],int W[], int V[], int n, int S)

{

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

{

for(int j = 0; j<= S; j++)

{

if(i == 0 || j == 0)

{

Value[i][j] = 0;

continue;

}

if(j< W[i])

{

Value[i][j] = Value[i-1][j];

continue;

}

int x = j - W[i];

if(Value[i-1][j] > V[i] + Value[i-1][x])

Value[i][j] = Value[i-1][j];

else

Value[i][j] = V[i] + Value[i-1][x];

}

}

}

void truyVet(int Value[50][100],int W[], int V[], int n, int S)

{

cout<< "Cac vat nen chon la: ";

int j = S;

for(int i = n; i>0; i--)

{

if(Value[i][j] != Value[i-1][j])

{

cout<< i<<" \t";

j -= W[i];

}

if(Value[i][j] == 0)

return;

}

}

int main()

{

int S,n;

cout << "Nhap tong khoi luong: "; cin >> S;

cout << "Nhap so do vat: "; cin>> n;

cout << endl;

int W[100], V[100];

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

{

cout<< "Nhap can nang [" << i <<"]:"; cin >> W[i];

cout<< "Nhap gia tri [" << i <<"]:"; cin >> V[i];

}

int Value[50][100];

createtable(Value,W,V,n,S);

xuat(Value,n,S);

cout << endl;

truyVet(Value,W,V,n,S);

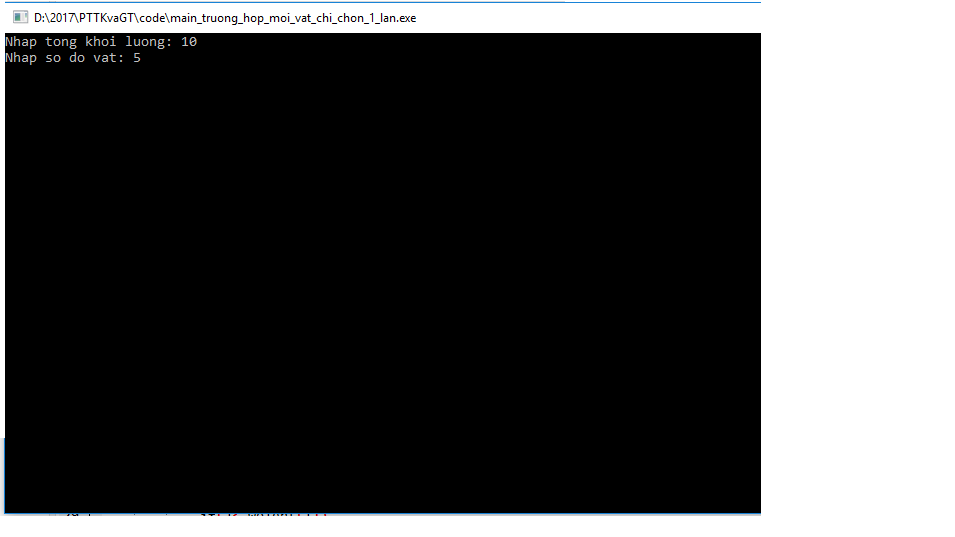
getch();

return 0;

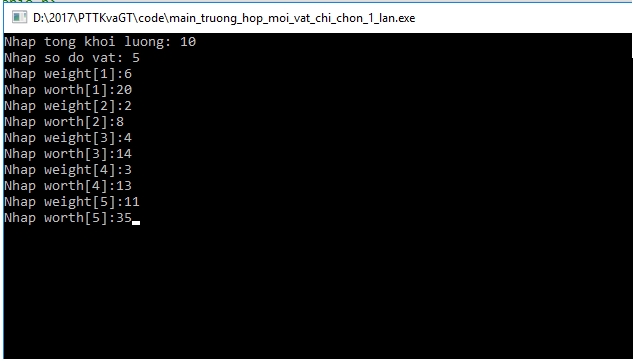
}

1. **Kết quả của chương trình**

* Nhập số đồ vật; khối lượng túi có thể đựng.

****

* Nhập các giá trị và khối lượng vật



* Kết quả là đồ vật 4, 3, 2 được chọn cho vào túi vì chúng có tổng giá trị lớn nhất.

