Lời mở đầu

Cùng với sự phát triển của khoa học kĩ thuật, công nghệ thông tin nói chung và bộ môn phân tích và thiết kế thuật toán nói riêng ngày càng được ứng dụng rộng rãi trong nhiều lĩnh vực. Với một cơ sở dữ liệu khổng lồ, việc đưa ra một phương pháp nhằm giải quyết vấn đề tìm kiếm dữ liệu có hiệu quả và nhanh chóng nhất luôn được sự quan tâm của các nhà phát triển phần mềm. Thông thường có rất nhiều phương pháp để giải quyết một bài toán. Việc truy suất dữ liệu chưa đạt hiệu quả cao. Sử dụng phương pháp quy hoạch động là một giải pháp làm tăng hiệu suất trong các thao tác xử lý.

Vấn đề đặt ra : để giải bài toán cái túi, chúng ta cần dùng phương pháp nào để đạt hiệu quả cao nhất. Để giải quyết vấn đề trên ta cùng tìm hiểu phương pháp quy hoạch động.

I. CƠ SỞ LÝ THUYẾT

1. Khái niệm

- Quy hoạch động là một phương pháp giảm thời gian chạy của các thuật toán thể hiện các tính chất của các bài toán con gối nhau (overlapping subproblem) và cấu trúc con tối ưu (optimal substructure).

2. Cách tiếp cận

- Top-down (Từ trên xuống): Bài toán được chia thành các bài toán con, các bài toán con này được giải và lời giải được ghi nhớ để phòng trường hợp cần dùng lại chúng. Đây là đệ quy và lưu trữ được kết hợp với nhau.
- Bottom-up (Từ dưới lên): Tất cả các bài toán con có thể cần đến đều được giải trước, sau đó được dùng để xây dựng lời giải cho các bài toán lớn hơn. Cách tiếp cận này hơi tốt hơn về không gian bộ nhớ dùng cho ngăn xếp và số lời gọi hàm. Tuy nhiên, đôi khi việc xác định tất cả các bài toán con cần thiết cho việc giải quyết bài toán cho trước không được trực giác lắm.

3. Các bước giải một bài toán với cấu trúc con tối ưu

- Chia bài toán thành các bài toán con nhỏ hơn.
- Giải các bài toán này một cách tối ưu bằng cách sử dụng đệ quy.
- Sử dụng các kết quả tối ưu xây dựng một lời giải tối ưu cho bài toán ban đầu.

4. Các bước giải một bài toán quy hoạch động

- Tên và ý nghĩa các biến phục vụ sơ đồ lặp.
- Cách khai báo các biến đó.
- Sơ đồ (công thức) lặp chuyển từ một bước sang bước tiếp theo.
- Giá trị đầu của các biến tham gia tính lặp.
- Tham số điều khiển lặp: thay đổi từ đâu đến đâu.
- Kết quả: ở đâu và làm thế nào để dẫn xuất ra.

II. BÀI TOÁN CÁI TÚI

1. Mô hình bài toán

Bài toán xếp cái túi (hay là bài toán ba lô) là một bài toán tối ưu hóa tổ hợp. Bài toán được đặt tên từ vấn đề chọn những gì quan trọng có thể bỏ vừa vào trong một cái túi (với giới hạn khối lượng) để mang theo trong một chuyến đi. Các bài toán tương tự thường xuất hiện trong kinh doanh, toán tổ hợp, lý thuyết độ phức tạp tính toán, mật mã học và toán ứng dụng.

2. Xây dựng hướng giải

a. Nhập và xuất dữ liệu

- Chọn phương án khai báo biến toàn cục.
- Chọn cách nhập dữ liệu từ bàn phím và xuất bảng tính ra màn hình.

b. Xây dựng bảng tính bằng phương pháp qui hoạch động

- Hàm mục tiêu f: tổng giá trị của cái túi (vali).
- Nhận xét: giá trị của cái túi phụ thuộc vào hai yếu tố, đó là giá trị của cái túi và trọng lượng của các đồ vật. Do đó ta có thể dùng mảng hai chiều để lưu trữ. F[i][j]: là tổng giá trị lớn nhất của cái túi khi xét từ vật thứ 1 đến vật thứ i và trọng lượng không vượt quá j.
- Khi xét đến f[i][j] thì các giá trị trên bảng phương án đều đượ tối ưu.
- Tính f[i][j] có 3 khả năng xảy ra:
  - Nếu f[i][0] = 0 và f[0][j] = 0.
  - Nếu a[i] > j thì f[i][j]=f[i-1][j].
  - Nếu a[i] <= j thì f[i][j]= max (f[i-1][j],f[i-1][j-a[i]] + c[i]).

c. Xây dựng hàm tìm giá trị lớn nhất

- Xây dựng hàm bằng cách so sáng hai giá trị (hai số) và đưa ra giá trị lớn hơn (số lớn hơn).

d. Xây dựng hàm truy vết tìm ra kết quả

- Xét từ cuối bảng:
  - Nếu f[i][j] != f[i-1][j] thì xuất giá trị đó ra.

III. CHƯƠNG TRÌNH BÀI TOÁN CÁI TÚI SỬ DỤNG

PHƯƠNG PHÁP QUI HOẠCH ĐỘNG

1. Chương trình

```
#include"stdio.h"
#include"conio.h"
int a[100],W, c[100], f[100][100];
int n,i,j,GT;
// nhap du lieu dau vao
void nhap( ){
        printf("\nNhap so luong do vat = "); scanf("%d", &n);
        printf("\nNhap khoi luong gioi han do vat = ");
        scanf("%d", &W);
    for( int i = 1; i<= n; i++ ){
         printf("\nNhap khoi luong do vat thu %d = ",i);
          scanf("%d", &a[i]);
        }
        for( int i = 1; i<= n; i++ ){
         printf("\nNhap vao so cong dung cua do vat thu %d = ",i);
         scanf("%d", &c[i]);
        }
}
// xuat bang tinh
void xuat(){
        printf("\n\n            **** BANG TINH****\n");
        for(i=1;i<=n;i++){
```

```
            for(j=0;j<=W;j++){
                    printf("%5d", f[i][j]);
            }
                    printf("\n");
            }
}
// tìm gia tri lon nhat
int max(int a, int b){
        return (a>b)?a:b;
}
// hàm tinh gia tri cua bang
int bangphuongan(){
    for(i=0;i<=n;i++){
        f[i][0]=0;
     }
    for(j=0;j<=W;j++){
        f[0][j]=0;
     }
    for(i=1;i<=n;i++){
     for(j=1;j<=W;j++){
      if (a[i]<=j){
        f[i][j]=max(f[i-1][j],f[i-1][j-a[i]]+c[i]);
   }
   else{
    f[i][j]=f[i-1][j];
          }
   }
}
}
// hàm tìm ket qua cua bai toan
```

```
int truyvet(){
        i=n;
        j=W;
        while ((i!=0)&&(j!=0)){
        if (f[i][j]!=f[i-1][j]){
        printf("%2d ",i);
        GT+=c[i];
        j-=a[i];
        }
        i--;
  }
}
int main(){
        nhap();
        printf("\n ****** CAC GIA TRI SAU KHI NHAP*****");
        printf("\n Trong luong gioi han cua tui la = %d\n",W);
        printf("\n trong luong cua do vat : \n");
          for(i=1;i<=n;i++){
            printf("%4d", c[i]);
        }
        printf("\n gia tri cua do vat : \n");
          for(i=1;i<=n;i++){
         printf("%4d", a[i]);
        }
        bangphuongan();
        xuat();
        printf("\n\n Cac do vat duoc cho vao tui la: ", i);
        truyvet();
        printf("\n\n Tong gia tri toi da cua tui la = %d",W);
    printf("\n\n Tong trong luong cua do vat duoc cho vao tui la= %d", GT);
```

```
getch();
return 0; }
```

2. Yêu cầu khi chạy

- Xuất ra được bảng tính.
- Hàm tìm kiếm có thể tìm ra được kết quả.

IV. DEMO CỦA BÀI TOÁN CÁI TÚI

1. Dữ liệu đầu vào

Trọng lượng giới hạn		W = 10
Số đồ vật giới hạn		n = 4
Giá trị / Đồ vật	**Trọng lượng của đồ vật**	**Công dụng đồ vật**
Đồ vật thứ 1	7	9
Đồ vật thứ 2	4	4
Đồ vật thứ 3	3	3
Đồ vật thứ 4	2	1

2. Kết quả

a. Bảng tính sử dụng phương pháp QHĐ

0	0	0	0	0	0	0	9	9	9	9
0	0	0	0	4	4	4	9	9	9	9
0	0	0	3	4	4	4	9	9	9	12
0	0	1	3	4	4	5	9	9	10	12

b. Kết quả bài toán

Giá trị được in / Giá trị cần in	**Giá trị in ra màn hình**
Tổng giá trị tối đa có thể cho vào túi.	10
Đồ vật được cho vào túi là đồ vật thứ.	3 1

Tổng công dụng (trọng lượng) của các đồ vật được cho vào túi.	12

V. KẾT LUẬN

Sau một thời gian tìm hiểu, nghiên cứu và thực hiện đề tài. Các yêu cầu chính của đề tài cơ bản đã hoàn tất với các nội dung sau:

1. Ưu điểm

- Xây dựng được chương trình " bài toán cái túi" sử dụng phương pháp qui hoạch động để giải.
- Chương trình sử lý nhanh và tương đối chính xác.

2. Khuyết điểm

- Mặc dù rất cố gắng nhưng trong thời gian ngắn, kinh nghiệm còn hạn chế nên kết quả còn thiếu sót cần tiếp tục được hoàn thiện để có thể giải được các yêu cầu phức tạp hơn.
- Chương trình còn nhiều lỗi như: về vấn đề xử lý hay thuật toán truy vết (tìm kiếm kết quả) chưa tối ưu …

3. Hướng phát triển

- Xây dựng hoàn thiện các chức năng giúp người sử dụng dễ dàng hơn, phương pháp qui hoạch động tương đối tối ưu và hiệu quả hơn.
- Có thể sử dụng phương pháp để giải một số bài toán tương tự.

Trên đây là kết quả đạt được cũng như còn một số tồn tại, hướng phát triển của đề tài.

Sinh viên thực hiện.

Đỗ Viết Vũ

TÀI LIỆU THAM KHẢO

1. Cẩm nang thuật toán – cuốn 1 – Robert Sedgewich – Trần Đan Thư.
2. Lập trình = Thuật toán + CTDL, N. Wirth.