

BÀI THỰC HÀNH SỐ 5: CHIẾN LƯỢC QUY HOẠCH ĐỘNG

Bài 1: Bài toán cái túi

Cho danh sách n gói hàng được đánh số thứ tự từ 1 đến n , gói hàng thứ i có trọng lượng là w_i và giá trị là v_i . Cài đặt chương trình thực hiện.

- Khởi tạo danh sách gồm tối thiểu 6 gói hàng.
- Sử dụng chiến lược quy hoạch động thiết kế thuật toán lấy các gói hàng trong số n gói hàng đã cho xếp vào một chiếc túi có thể chứa được khối lượng tối đa là m (m nhập từ bàn phím) sao cho giá trị đạt được là lớn nhất.
- Hiển thị ra màn hình các gói hàng đã lấy, mỗi gói hàng hiển thị số thứ tự, khối lượng và giá trị và tổng số gói đã lấy.

Bài 2: Phân tích số thành tổng

Cho 2 số tự nhiên n và m

Cài đặt chương trình thực hiện.

- Khởi tạo các số n và m .
- Sử dụng chiến lược quy hoạch động thiết kế thuật toán tính số cách phân tích số n thành tổng các số nguyên dương nhỏ hơn hoặc bằng m .
- Sử dụng phương pháp sinh hoặc phương pháp quay lui để liệt kê (và hiển thị) các cấu hình.

Bài 3: Dãy con đơn điệu tăng dài nhất.

Cài đặt chương trình giải quyết bài toán dãy con đơn điệu tăng dài nhất.

- Khởi tạo dãy gồm tối thiểu 15 số thực.
- Cài đặt thuật toán tìm dãy con đơn điệu tăng dài nhất theo chiến lược quy hoạch động.
- Hiển thị kết quả: độ dài của dãy con và giá trị các phần tử của dãy con.

Bài 4: Đổi tiền.

Một ngân hàng có n loại tiền mệnh giá $c[1], c[2], \dots, c[n]$ với số lượng tờ tiền mỗi loại không giới hạn. Ngân hàng cần chi trả cho khách hàng một số tiền m đồng. Hãy cho biết cần bao nhiêu tờ tiền mỗi loại để chi trả sao cho số lượng tờ tiền phải trả là ít nhất.

Cài đặt chương trình thực hiện:

- Khởi tạo số n , mảng c và số m .
- Cài đặt thuật toán theo chiến lược quy hoạch động.
- Hiển thị kết quả gồm tổng số tờ tiền được chi trả và các loại mệnh giá được chi trả.

Hướng dẫn:

Gọi $f(i)$ là số lượng tờ ít nhất để trả số tiền i nên $f(m)$ là số tờ ít nhất để trả số tiền m .

Để được số tiền là i có các cách để tạo thành số tiền đó khi chúng ta dùng thêm một tờ là: $i - c[k_1]$, $i - c[k_2]$, ..., $i - c[k_j]$, trong đó k_j là số thỏa mãn mà $c[k_j] < i$.

Số tờ tiền tối ưu nhất cần tìm trong các $f(i - c[k_1]) + 1$, $f(i - c[k_2]) + 1$, ..., $f(i - c[k_j]) + 1$.

Công thức quy hoạch động như sau:

$$f(i) = \min\{f(i - c[j]) + 1, j \text{ thỏa mãn: } c[j] < i\}$$

Thuật toán quy hoạch động:

```
coin_changing() {
    f[0] = 0;
    for (i = 1 -> m) {
        min = maxint;
        for (j = 1 -> n) {
            if (f[i - a[j]] + 1 < min && a[j] < i) {
                min = f[i - a[j]] + 1;
                s[i] = j;
            }
        }
        f[i] = min;
    }
}
```

Giải thích:

S là mảng chứa đựng là loại tiền nào cần dùng cuối cùng để đạt số tiền i .

Truy vết để tìm lại các loại tiền cần dùng bằng mảng s như sau:

```
j = s[m];
i = m;
while (j khác 0) {
    print(a[j]);
    i = i - j;
    j = s[i];
}
```