BÀI 5. GIẢI THUẬT QUY HOẠCH ĐỘNG

NỘI DUNG

1. Mô tả thuật toán
2. Đánh giá thuật toán
3. Các bài toán áp dụng

Thuật toán Quy hoạch động (Dynamic Programming)

Phương pháp quy hoạch động dùng để giải lớp các bài toán thỏa mãn những điều kiện sau:

1. Bài toán lớn cần giải có thể phân rã được thành nhiều bài toán con.
2. Sự phối hợp lời giải của các bài toán con cho ta lời giải của bài toán lớn:
   - Bài toán con có lời giải đơn giản được gọi là **cơ sở** của quy hoạch động.
   - Công thức phối hợp nghiệm của các bài toán con gọi là **công thức truy hồi**
3. Có không gian vật lý lưu trữ lời giải các bài toán con (**Bảng phương án** của quy hoạch động).
4. Quá trình giải quyết từ bài toán cơ sở (bài toán con) để tìm ra lời giải bài toán lớn phải được thực hiện sau hữu hạn bước dựa trên bảng phương án của quy hoạch động.

SO SÁNH QUY HOẠCH ĐỘNG VỚI CHIA VÀ TRỊ

- **Trong giải thuật chia và trị:**
  - Các bài toán con độc lập, sau đó các bài toán con này được giải một cách đệ quy.
- **Trong giải thuật quy hoạch động:**
  - Các bài toán con là không độc lập với nhau, mà cùng có chung các bài toán con nhỏ hơn.

Các yếu tố của giải thuật Quy hoạch động

Cơ sở của quy hoạch động:

- Những trường hợp đơn giản có thể tính trực tiếp

Cấu trúc con tối ưu:

- Phương pháp chia nhỏ các bài toán cho đến khi gặp được bài toán cơ sở.

Tổng hợp:

- Hệ thức truy hồi tính giá trị tối ưu của hàm mục tiêu của bài toán lớn qua giá trị tối ưu của các bài toán con thành phần.

NHẬN XÉT VỀ QUY HOẠCH ĐỘNG

Ưu điểm:

- Độ phức tạp tốt.

Nhược điểm:

- Chỉ áp dụng cho các bài toán tối ưu nhưng không cần biết đầy đủ các phương án tối ưu, chỉ quan tâm đến kết quả tối ưu
- Bảng phương án sẽ tốn bộ nhớ

Ví dụ: Tính số Fibonacci thứ n

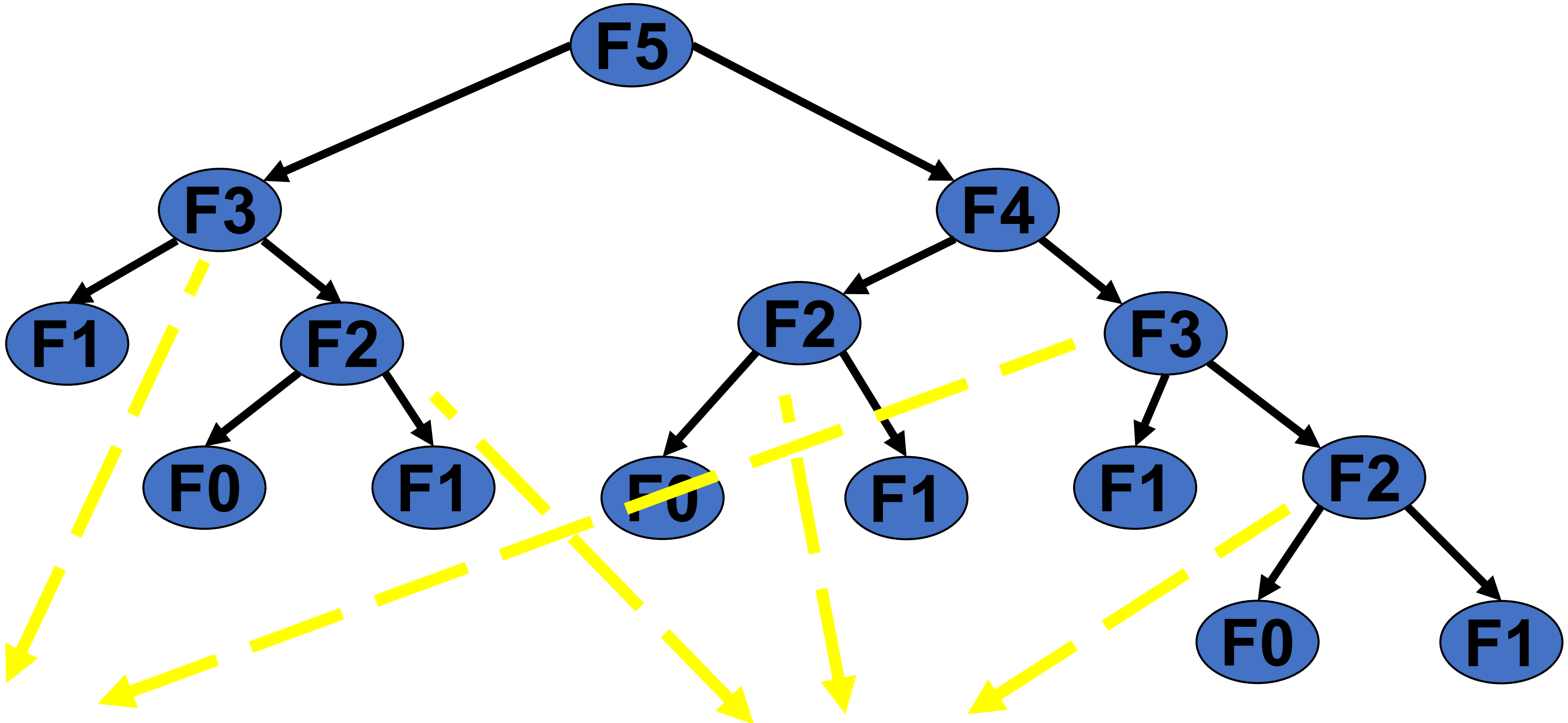
Định nghĩa số Fibonacci F(n):

- F(0)=0
- F(1)=1
- F(n)=F(n-2)+F(n-1) với n>1

Ví dụ:

F(2)=1, F(3)= 2, F(4) = 3 , F(5)=5, F(6)=8

Tính số Fibonacci kiểu đệ quy



- 2 lần tính F(3)
- 3 lần tính F(2)

Thuật toán Quy hoạch động cho dãy Fibonacci

- **Bước cơ sở** : Tính F[1] =1, F[2] =1.

- **Công thức truy hồi:** Lưu lời giải các bài toán con biết trước vào bảng phương án (*ở đây là mảng một chiều F[100]*). Sử dụng lời giải của bài toán con trước để tìm lời giải của bài toán con tiếp theo.
  - Trong bài toán này là F[i] = F[i-1] + F[i-2] với $n \geq 3$.
  - Bằng cách lưu trữ vào bảng phương án, ta chỉ cần giải mỗi bài toán con một lần.

- **Truy vết**: Đưa ra nghiệm của bài toán bằng việc truy lại dấu vết phối hợp lời giải các bài toán con để có được nghiệm của bài toán lớn.

Thuật toán Quy hoạch động cho dãy Fibonacci

```
Procedure Fibonacci(N)
    F[1] = 1
    F[2] = 1
    For i:=3 to N do
          F[i]: = F[i-1] + F[i-2]
```

Độ phức tạp: O(N)

CÁC BÀI TOÁN ÁP DỤNG QUY HOẠCH ĐỘNG

1. Cái túi 0-1
2. Dãy con chung dài nhất
3. Dãy con liên tiếp có tổng lớn nhất
4. Dãy con tăng dài nhất
5. Dãy con có tổng bằng S
6. Xâu con đối xứng dài nhất
7. Tính tổ hợp

1. Bài toán cái túi (dạng 0-1)

Bài toán

- Có N gói đồ vật, gói thứ i có khối lượng là w[i], có giá trị là v[i]
- Cái túi chỉ có thể mang được khối lượng tối đa là **M**.
- Trong bài toán cái túi dạng 0–1, mỗi gói đồ vật chỉ có thể lấy nguyên vẹn hoặc không lấy.

1. Bài toán cái túi (dạng 0-1)

➢Gọi C[i, L] là giá trị lớn nhất đạt được khi xét i đồ vật từ 1 đến i với kích thước cái túi là L.

➢Hai trường hợp

- **Bài toán con 1**:
  - Nếu chọn vật thứ i (nếu w[i] ≤ L), giá trị lớn nhất có thể là: **C(i–1, L– w[i]) + v[i]** ;
- **Bài toán con 2**:
  - Nếu không chọn vật thứ i, giá trị lớn nhất là **C(i–1, L)**

Công thức truy hồi

- C(i, L) =max{C(i –1,L– w[i]) +v[i] , C(i –1,L)}

Giải thuật QHĐ cho bài toán cái túi

{Khởi tạo}: For L: = 0 to M do C[0,L] :=0 ;

{Lặp:}

```
For i = 1 to N do
  For L = 1 to M do
    Begin
          C[i,L] := C[ i–1,L];
          If  (w[i] ≤ L) and  (C[i–1,L–w[i]] + v[i] > C[i-1, L]) then
                              C[i, L] := C[i–1,L–w[i]]+v[i] ;
    End;
Return C(N, M)
```

2. Bài toán dãy con chung dài nhất

Bài toán

- Cho hai dãy $X = (x_1,x_2,\ldots,x_m)$ và $Y = (y_1,y_2,\ldots,y_n)$ gồm các số nguyên.
- Cần tìm dãy con chung dài nhất của hai dãy X và Y.

- Bài toán tương tự: Cho hai xâu ký tự X độ dài m và Y độ dài n. Hãy tìm xâu con chung dài nhất.

2. Bài toán dãy con chung dài nhất

Bài toán con cơ sở

$C[0, j] = 0 \; \forall j = 0..n$ và $C[i,0] = 0, i = 0..m$.

(là độ dài dãy con chung dài nhất của dãy rỗng với một dãy khác).

TỔNG HỢP Với $i > 0, j > 0$. Tính $C[i, j]$. Có hai tình huống:

- Nếu $x_i = y_j$ thì dãy con chung dài nhất của X_i và Y_i => bổ sung x_i vào dãy con chung dài nhất của hai dãy X_{i-1} và Y_{j-1}
- Nếu $x_i \neq y_i$: dãy con chung dài nhất của X_i và Y_j sẽ là dãy con dài hơn trong hai dãy con chung dài nhất của (X_{i-1} và Y_i) và của (X_i và Y_{j-1}) .

2. Bài toán dãy con chung dài nhất

Công thức truy hồi:

- C[i,j] = 0 nếu i =0 hoặc j=0
- C[i,j] = C[i-1,j-1]+1 nếu $x_i = y_j$
- C[i,j] = Max{ C[i-1,j], C[i,j-1]} nếu $x_i \neq y_j$

3. Bài toán dãy con liên tiếp có tổng lớn nhất

Cho dãy A dưới dạng mảng A[1..n] các số nguyên, cả âm và dương.

Hãy tìm dãy con các phần tử liên tiếp của dãy A có tổng lớn nhất

Kết quả: In ra tổng lớn nhất và dãy con tương ứng.

3. Bài toán dãy con liên tiếp có tổng lớn nhất

Gọi S(i) là tổng của dãy con lớn nhất trong dãy i phần tử

A_i = a[1], …., a [i], i = 1,2,…, n

S(n) là giá trị cần tìm.

Bài toán con cơ sở

Với i =1 ta có S(1)= a[1].

3. Bài toán dãy con liên tiếp có tổng lớn nhất

- Gọi E(i) là tổng lớn nhất của các dãy con liên tiếp của dãy a[1]..a[i] chứa chính a[i].

- Xét một trong hai trường hợp:
  - Các dãy con liên tiếp có chứa a[i] => Tổng lớn nhất là S(i-1)
  - Các dãy con liên tiếp không chứa a[i] => Tổng lớn nhất là E(i)

- Tổng hợp: **S(i) = max {S(i–1) , E(i)}.**

3. Bài toán dãy con liên tiếp có tổng lớn nhất

Để tính E(i), i = 1, 2, …, n:

1. Với i=1: E(i) = a[1];

2. Với i >1, có hai khả năng:
   - Nếu dãy chứa a[i–1], độ dài lớn nhất có thể là E(i–1)+a[i]. Xảy ra nếu E(i-1) là số dương.
   - Nếu dãy không chứa a[i–1] thì dãy chỉ có a[i]. Xảy ra khi E(i-1) là số âm.

3. *Tổng hợp:* **E[i] = max {a[i] , E[i–1] + a[i] }, i > 1.**

4. Dãy con tăng dài nhất

- Cho dãy số A có N phần tử, bài toán yêu cầu tìm dãy con dài nhất của dãy A sao cho phần tử sau của dãy con luôn lớn hơn phần tử trước.
- Dãy con của một dãy số là dãy có được sau khi loại bớt một số phần tử, các phần tử khác giữ nguyên vị trí.
- Dãy con tăng của A là một dãy $A(i_1),A(i_2),...,A(i_k)$ thỏa mãn

$$i_1<i_2<...<i_k \quad \text{và} \quad A(i_1)<A(i_2)<...<A(i_k)$$

4. Dãy con tăng dài nhất

Gọi F(i) là độ dài dãy con tăng dài nhất kết thúc ở A(i), ta có công thức tính:

- F(1)=1
- F(i)=max{F(j)+1}

  Với j,j thỏa mãn 1≤j<ivà A(j)<A(i)

Kết quả bài toán là **max{F}**

5. DÃY CON CÓ TỔNG BẰNG S

Bài toán:

- Cho dãy $A_1,A_2,...,A_N$. Xác định **có hay không** một dãy con của dãy đó có tổng bằng S.

Giải pháp:

- Sử dụng bảng phương án L là một ma trận nhị phân
- Đặt L[i,t]=1 nếu có thể tạo ra tổng t từ một dãy con của dãy gồm các phần tử A1,A2,...,Ai.
- Ngược lại thì L[i,t]=0. Nếu L[n,S]=1 thì đáp án của bài toán trên là “có”.

5. DÃY CON CÓ TỔNG BẰNG S

Ta có thể tính L[i,t] theo công thức:

- L[i,t]=1 nếu L[i−1,t]=1
- hoặc L[i−1,t−a[i]]=1.

Nhận xét:

- Để tính dòng thứ i, ta chỉ cần dòng i−1.
- Bảng phương án khi đó chỉ cần 1 mảng 1 chiều L[0..S]

5. DÃY CON CÓ TỔNG BẰNG S

```
L[t]:=0; L[0]:=1;

for i := 1 to n do
    for t := S downto a[i] do
        if (L[t]=0) and (L[t-a[i]]=1) then
                      L[t]:=1;
```

6. XÂU CON ĐỐI XỨNG DÀI NHẤT

Bài toán: Cho một xâu S, độ dài không quá 1000 kí tự. Tìm xâu đối xứng dài nhất là xâu con của S (Xâu con là một dãy các kí tự liên tiếp).

Giải pháp: Dùng ma trận F[i, j] có ý nghĩa: F[i, j] = true/false nếu đoạn gồm các kí tự từ i đến j của S có/không là xâu đối xứng.

Công thức:

1. F[i, i] = True: xâu 1 ký tự luôn đối xứng.
2. F[i, j] = F[i+1, j-1] nếu Si=Sj.
3. F[i, j] = False nếu Si≠Sj.

6. XÂU CON ĐỐI XỨNG DÀI NHẤT

```
for i := 1 to n do
    F[i, i] := True;
for k := 1 to (n-1) do
    for i := 1 to (n-k) do
        begin
             j := i + k;
             F[i, j] := ( F[i+1, j-1] ) and (s[i] = s[j]
);
        end;
```

Kết quả: Max(j−i+1)≤j thỏa mãn F[i,j]=True.

Độ phức tạp thuật toán là $O(N^2)$.

7. Tính tổ hợp C(n,k)

Bài toán: Tính C(n,k) với n đến 1000.

Phương pháp quy hoạch động: Áp dụng công thức

$$C_n^0 = C_n^n = 1$$

$$C_n^k = C_{n-1}^k + C_{n-1}^{k-1}$$

Chú ý: Vì giá trị sẽ rất lớn nên cần chia dư sau mỗi bước tính.

7. Tính tổ hợp C(n,k)

Cài đặt:

```
for (i = 0; i <= 1000; i++)    {
        for (j = 0; j <= i; j++)    {
            if (j == 0 || j == i)
                C[i][j] = 1;
            else
                C[i][j] = (C[i-1][j-1] + C[i-1][j]) % MOD;
        }
    }
```