CHƯƠNG 5
CHIẾN LƯỢC QUY HOẠCH ĐỘNG

Nội dung chính chương 5

5.1 Chiến lược quy hoạch động

5.1.1 Bài toán Dãy Fibonacci

5.1.2 Bài toán nhân dãy các ma trận

5.2 Phương pháp quy hoạch động

5.3 Bài toán dãy con chung dài nhất

5.1 Chiến lược quy hoạch động

- QHĐ là một PP chung rất hiệu quả để giải quyết các bài toán tối ưu.
- Tìm cấu hình thoả mãn tất cả ràng buộc và đạt giá trị tốt nhất theo một mục tiêu nào đó.
- Dùng để giải bài toán tối ưu có bản chất đệ quy, có thể đưa về tìm phương án tối ưu của một số hữu hạn các bài toán con.

5.1 Chiến lược quy hoạch động

- **Chia để trị**: Để giải quyết bài toán lớn, ta chia nó làm nhiều bài toán con cùng dạng với nó để có thể giải quyết độc lập (từ trên xuống).
- **Quy hoạch động**: Giải quyết tất cả các bài toán con và lưu trữ những lời giải hay đáp số của chúng nhằm phối hợp lại để giải quyết những bài toán tổng quát hơn (từ dưới lên).

5.1 Chiến lược quy hoạch động

Ưu điểm:

- Thực hiện nhanh do không phải tốn thời gian giải lại một bài toán con đã được giải.
- Vận dụng để giải các bài toán tối ưu, các bài toán có công thức truy hồi.

Nhược điểm: không hiệu quả khi

- Không tìm được công thức truy hồi.
- Số lượng các bài toán con cần giải quyết và lưu giữ kết quả là rất lớn.
- Sự kết hợp lời giải của các bài toán con chưa chắc cho ta lời giải của bài toán ban đầu.

5.1.1 Bài toán Dãy Fibonacci

- Bài toán: Dãy Fibonacci là dãy vô hạn các số nguyên dương F[1], F[2],… được định nghĩa như sau

$$F[i] = \begin{cases} 1, & \text{if } i \le 2 \\ F[i-1] + F[i-2], & \text{if } i \ge 3 \end{cases}$$

5.1.1 Bài toán Dãy Fibonacci

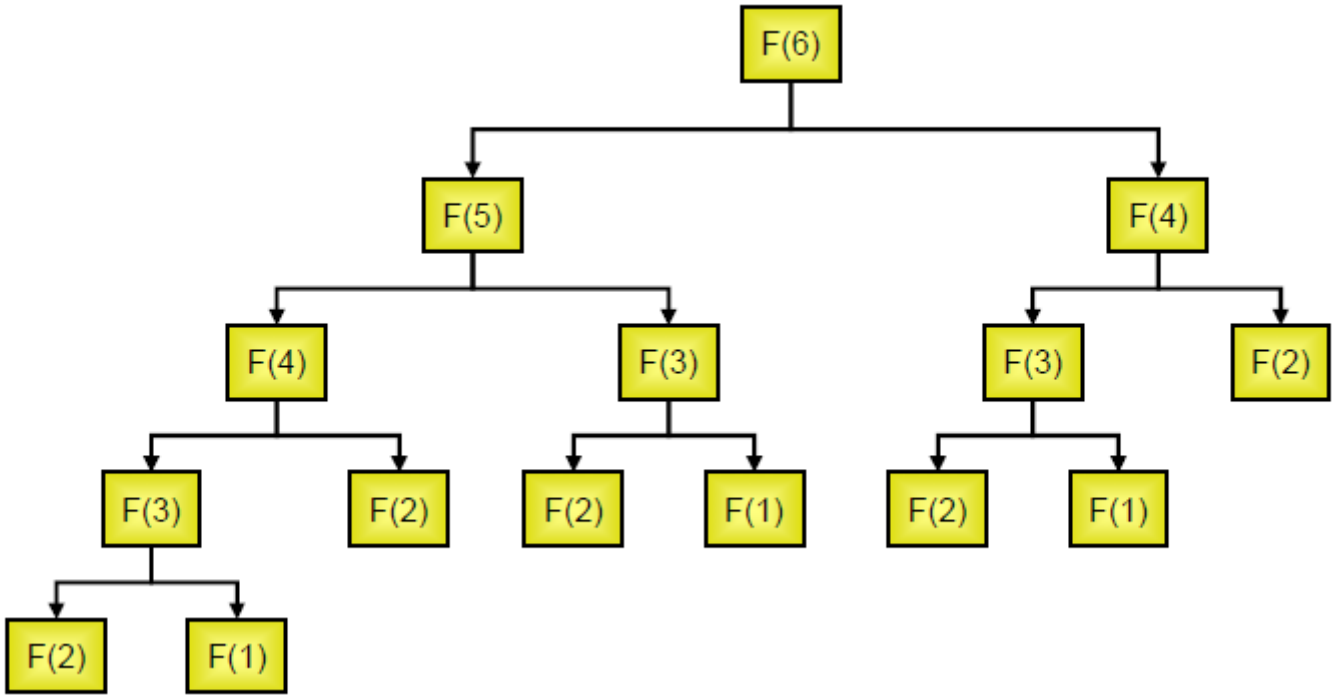
- Giải thuật đệ quy

```
function F(i: Integer): Integer;
begin
  if i < 3 then F := 1
  else F := F(i - 1) + F(i - 2);
end;
```

- Độ phức tạp: $O(2^n)$

5.1.1 Bài toán Dãy Fibonacci

- Hàm đệ quy tính Fibonacci F(6)



- Nhận xét: Phải tính lại nhiều lần các bài toán con.

5.1.1 Bài toán Dãy Fibonacci

- Cải tiến: Lưu lại kết quả các bài toán con đã tính để dùng lại.
- Giải thuật

```
begin
   F[1] := 1; F[2] := 1;
   for i := 3 to n do
      F[i] := F[i-1] + F[i-2];
   WriteLn(F[n]);
end.
```

- Độ phức tạp: O(n).
- QHĐ tính toán đệ quy hiệu quả bằng cách lưu trữ các kết quả cục bộ.
  Kết quả các bài toán con thường được lưu vào mảng.

5.1.2 Bài toán nhân dãy các ma trận

- Bài toán:

  Nhân một dãy các ma trận : A x B x C x D x…sao cho số phép nhân ít nhất.

Ví dụ: A_{30x1}, B_{1x40}, C_{40x10}, D_{10x25}

- ((AB)C)D = 30x1x40 + 30x40x10 + 30x10x25 = 20700
- (AB)(CD) = 30x1x10 + 40x10x25 + 30x40x25 = 41200
- A((BC)D) = 1x40x10 +1x10x25 + 30x1x25 =1400

5.1.2 Bài toán nhân dãy các ma trận

Ta sử dụng tính kết hợp của nhân ma trận và chiến lược QHĐ để tìm kết quả tối ưu: số phép nhân ít nhất.

- Gọi M(i,j): số phép nhân nhỏ nhất để tính $\prod_{k=i}^{j} A_k$
- Các dấu ngoặc ngoài cùng phân hoạch dãy các ma trận (i,j) tại một vị trí k nào đó.
- Cả hai nửa của dãy các ma trận (i,j) tại điểm phân hoạch k sẽ đều có thứ tự các dấu là tối ưu.

5.1.2 Bài toán nhân dãy các ma trận

- Từ đó có công thức truy hồi
  - M(i,j) = 0 nếu i=j
  - $M(i,j) = \min_{i \le k \le j-1} [M(i,k) + M(k+1,j) + d_{i-1} d_k d_j]$
  - Trong đó ma trận A_i có kích thước là (d_{i-1}, d_i).

M(1,n) là kết quả tối ưu cần tìm.

Các cặp bài toán con (i,j) khác nhau: $O(n^2)$.

Do đó, cần một mảng có kích thước $O(n^2)$.

Mỗi bài toán con yêu cầu tính toán tối đa : O(n)

Vậy, độ phức tạp về thời gian của giải pháp QHĐ: $O(n^3)$.

5.1.2 Bài toán nhân dãy các ma trận

```
for i:=1 to n do
  F[i,i]:=0;

for i:=1 to n-1 do
  F[i,i+1] := d[i-1]*d[i]*d[i+1];

for m:=2 to n-1 do
  begin
    for i:=1 to n-m do
      begin
        j:=i+m;
        F[i,j]:=oo;
        for k:=i+1 to j-1 do
          F[i,j]:=min(F[i,j], F[i,k]+F[k+1,j]+d[i-1]*d[k]*d[j]);
      end;
end;
```

5.2 Phương pháp quy hoạch động

- B1: Xác định đặc điểm cấu trúc của giải pháp tối ưu của bài toán.
- B2: Tìm công thức truy hồi (đệ qui) xác định giá trị của một giải pháp tối ưu.
- B3: Tính giá trị tối ưu của bài toán dựa vào các giá trị tối ưu của các bài toán con của nó (bottom-up).
- B4: Xây dựng nghiệm giá trị tối ưu từ các thông tin đã tính.

5.2 Phương pháp quy hoạch động

Điều kiện áp dụng QHĐ:

- Bài toán lớn phải phân rã được thành nhiều bài toán con, mà sự phối hợp lời giải của các bài toán con đó cho ta lời giải của bài toán lớn.
- Phải có đủ không gian vật lý lưu trữ lời giải (bộ nhớ, đĩa…) để phối hợp lời giải của các bài toán con.
- Quá trình từ bài toán cơ sở tìm ra lời giải bài toán ban đầu phải qua hữu hạn bước.

5.3 Bài toán dãy con chung dài nhất

Cho 2 dãy ký tự X [n] và Y[m] tìm dãy con chung lớn nhất của hai dãy trên.

Với dãy con của một dãy được định nghĩa là một tập con các ký tự của dãy(giữ nguyên thứ tự).

Ví dụ

X = A L G O R I T H M

Y = L O G A R I T H M

Thì dãy con chung dài nhất là:

Z = LORITHM

5.3 Bài toán dãy con chung dài nhất

B1: Xác định đặc điểm của dãy con chung dài nhất (giải pháp tối ưu của bài toán).

- Giả sử chúng ta đã có lời giải là Z[1..k].
- Nếu hai ký tự cuối cùng của X và Y trùng nhau thì đó cũng là ký tự cuối cùng của Z.
- Phần còn lại của Z khi đó sẽ là xâu con chung dài nhất của X[1..n-1] và Y[1..m-1].

5.3 Bài toán dãy con chung dài nhất

- Nếu hai ký tự cuối của X và Y không trùng nhau thì một trong số chúng sẽ không nằm trong Z (có thể cả hai).
- Giả sử ký tự không nằm trong Z trong trường hợp đó là ký tự của X.
- Thế thì Z sẽ là dãy con dài nhất của X[1..n-1] và Y[1..m].
- Ngược lại nếu ký tự không nằm trong Z là ký tự của Y thì Z sẽ là dãy con dài nhất của X[1..n] và Y[1..m-1].

5.3 Bài toán dãy con chung dài nhất

B2: Xây dựng công thức truy hồi tính độ dài lớn nhất của dãy con của 2 dãy.

Từ b1 xây dựng công thức truy hồi như sau:

- Gọi C[i][j] là độ dài dãy con lớn nhất của hai dãy X[1..i] và Y[1..j].
- C[i][0] = C[0][j] với mọi i, j.
- Lời giải của bài toán chính là C[n][m].
- $C[i][j] = \begin{cases} C[i-1][j-1] + 1 \text{ nếu } X[i] = Y[j] \\ \max(C[i-1][j], C[i][j-1] \text{ nếu } X[i] \neq Y[j] \end{cases}$

5.3 Bài toán dãy con chung dài nhất

		A	L	G	O	R	I	T	H	M
	0	0	0	0	0	0	0	0	0	0
L	0	0	1	1	1	1	1	1	1	1
O	0	0	1	1	2	2	2	2	2	2
G	0	0	1	2	2	2	2	2	2	2
A	0	1	1	2	2	2	2	2	2	2
R	0	1	1	2	2	3	3	3	3	3
I	0	1	1	2	2	3	4	4	4	4
T	0	1	1	2	2	3	4	5	5	5
H	0	1	1	2	2	3	4	5	6	6
M	0	1	1	2	2	3	4	5	6	7

5.3 Bài toán dãy con chung dài nhất

B3: Xây dựng thuật toán tìm dãy con chung dài nhất của 2 dãy X[1..n] và Y[1..m].

Thứ tự tính toán diễn ra như sau: ban đầu chúng ta tính C[1][1], C[1][2], ..., C[1][n],C[2][1], C[2][2], ..., C[2][n], ...

Độ phức tap để tính C [i][j] bằng O(1) với i=1,n và j=1,m nên độ phức tạp của thuật toán là O(mn).

5.3 Bài toán dãy con chung dài nhất

```
int longest_common_sequence(X, Y)
{
        for(i=0;i≤m;i++)
                C[i][0] = 0;
        for(j=0;j≤n;j++)
                C[0][j] = 0;
        for(i=1;i≤m;i++)
                for(j=1;j≤n;j++)
                        if(X[i]==Y[j])
                                C[i][j] = C[i-1][j-1] + 1;
                        else
                                C[i][j] = max(C[i-1][j],C[i][j-1]);
return C[m][n];
}
```

5.3 Bài toán dãy con chung dài nhất

		A	L	G	O	R	I	T	H	M
	0	0	0	0	0	0	0	0	0	0
L	0	0	1	1	1	1	1	1	1	1
O	0	0	1	1	2	2	2	2	2	2
G	0	0	1	2	2	2	2	2	2	2
A	0	1	1	2	2	2	2	2	2	2
R	0	1	1	2	2	3	3	3	3	3
I	0	1	1	2	2	3	4	4	4	4
T	0	1	1	2	2	3	4	5	5	5
H	0	1	1	2	2	3	4	5	6	6
M	0	1	1	2	2	3	4	5	6	7

5.3 Bài toán dãy con chung dài nhất

B4: Tìm dãy con dài nhất của X [1..n] và Y[1..m](Xây dựng nghiệm đạt giá trị tối ưu từ các thông tin đã tính).

- Để tìm lại được nghiệm chúng ta sử dụng một bảng D[i][j] trỏ ngược tới (i,j-1) hoặc (i-1,j) hoặc (i-1,j-1) và lần ngược từ D[m][n] như sau: nếu D[i][j] trỏ tới (i-1, j-1) thì X[i] = Y[j] là ký tự này sẽ nằm trong dãy con dài nhất của 2 xâu.
- Việc D[i][j] trỏ tới (i-1,j-1), (i-1,j) hoặc (i,j-1) phụ thuộc vào giá trị của mảng C tại vị trí nào được sử dụng để tính C[i][j].
- Các giá trị của mảng D sẽ được tính như sau: D[i][j] bằng 1 (trên trái) nếu C[i][j] = 1 + C[i-1][j-1], bằng 2 (trên) nếu C[i][j] = C[i- 1][j] và bằng 3 (trái) nếu C[i][j] = C[i][j-1].

5.3 Bài toán dãy con chung dài nhất

		A	L	G	O	R	I	T	H	M
	0	0	0	0	0	0	0	0	0	0
L	0	0	1	1	1	1	1	1	1	1
O	0	0	1	1	2	2	2	2	2	2
G	0	0	1	2	2	2	2	2	2	2
A	0	1	1	2	2	2	2	2	2	2
R	0	1	1	2	2	3	3	3	3	3
I	0	1	1	2	2	3	4	4	4	4
T	0	1	1	2	2	3	4	5	5	5
H	0	1	1	2	2	3	4	5	6	6
M	0	1	1	2	2	3	4	5	6	7

Thuật toán tìm nghiệm: Nếu D[i][j] trỏ tới (i-1, j-1) (1 – trên trái) thì X[i] = Y[j] và ký tự này sẽ nằm trong dãy con là nghiệm.

5.3 Bài toán dãy con chung dài nhất

```
char * findSolution()
{
row = m, col = n, lcs="";
while((row>0)&&(col>0))
{
if(D[row][col] == 1)
{
lcs = lcs + X[row];
row = row – 1;
col = col - 1;
}else{
if (D[row][col]==2)
row = row – 1;
else if (D[row][col] = 3)
col = col – 1;
}
}
reverse lcs; // đảo ngược xâu lcs
return lcs;
}
```

5.3 Bài toán dãy con chung dài nhất

		A	L	G	O	R	I	T	H	M
	0	0	0	0	0	0	0	0	0	0
L	0	0	1	1	1	1	1	1	1	1
O	0	0	1	1	2	2	2	2	2	2
G	0	0	1	2	2	2	2	2	2	2
A	0	1	1	2	2	2	2	2	2	2
R	0	1	1	2	2	3	3	3	3	3
I	0	1	1	2	2	3	4	4	4	4
T	0	1	1	2	2	3	4	5	5	5
H	0	1	1	2	2	3	4	5	6	6
M	0	1	1	2	2	3	4	5	6	7

Bài tập

1. Cho một hình chữ nhật kích thước 2xN ($1<=N<10^9$). Hãy đếm số cách lát các viên gạch nhỏ kích thước 1x2 và 2x2 vào hình trên sao cho không có phần nào của các viên gạch nhỏ thừa ra ngoài, cũng không có vùng diện tích nào của hình chữ nhật không được lát.

Input: Gồm nhiều test, dòng đầu ghi số lượng test T

(T<=10). T dòng sau mỗi dòng ghi một số N.

Output: Ghi ra T dòng là số cách lát tương ứng lấy phần dư cho 10^9+7

Bài tập

2. Xét dãy số Fibonacci $\{F_n\}$ theo định nghĩa:

$F_0=F_1=1$

$F_n=F_{n-1}+F_{n-2}\ \forall n>1$

Cho số n, hãy tính tổng $S=F_0+F_1+F_2+\cdots+F_n$ và đưa ra số dư của S chia cho (10^9+7)

Input: số n nguyên dương, $n\leq 10^{15}$

Output: số dư của S chia cho (10^9+7)

Bài tập

3. Bờm đang nghiên cứu mực nước biển ở hành tinh Quạt Mo. Sau nhiều ngày theo dõi, Bờm nhận thấy rằng quy luật của mực nước biển là: mực nước biển của một ngày bất kì bằng trung bình cộng mực nước biển của ngày hôm trước và ngày hôm sau. Dựa vào ghi chép mực nước biển hai ngày đầu của Bờm, hãy tính toán mực nước biển ngày thứ N

Input: Dòng 1: chứa 2 số nguyên b, a là mực nước biển 2 ngày đầu ($-100 \leq a, b \leq 100$). Số a là mực nước ngày thứ nhất, số b là mực nước ngày thứ 2. Dòng 2: chứa số nguyên dương N ($3 \leq N \leq 10^{12}$).

Output: mực nước biển ngày thứ N.

Bài tập

4. Turing hiện đang làm việc để crack các máy bí ẩn. Nhưng ông thấy rằng có hai hàm toán học là f(n) và g(n) được sử dụng để mã hóa tin nhắn của người Đức. Ông muốn thử nghiệm khám phá của mình để bắt chước cách mã hóa của máy tính. Các hàm được định nghĩa là:

g (n + 1) = 4 * g (n) + f (n + 1) + c

f (n + 2) = 3 * f (n + 1) + 2 * f (n)

Dữ liệu ban đầu: f (0) = 1; f (1) = 1; g (-1) = 1; g (0) = 1; c= 2

Yêu cầu: Cho số nguyên n, cần phải tìm giá trị g(n) modulo 10^8+7.

Đầu vào: Dòng đầu tiên ghi số lượng các trường hợp thử nghiệm T, T dòng tiếp theo mỗi dòng ghi một giá trị của n.

Đầu ra: Với mỗi test, xuất ra giá trị của g(n).

$1 \leq T \leq 1000$

$1 \leq n \leq 10^7$


Thank You!