QUY HOẠCH ĐỘNG 3

Quy hoạch động

1. **Bài toán ba lô (knapsack 0-1)**
2. **Nguyên lý tối ưu Belman**
3. **Đánh giá thuật toán**
4. **Bài toán dãy con chung dài nhất**
5. **Bài toán dãy con đơn diệu dài nhất**
6. **Bài tập**

5. Dãy con đơn điệu dài nhất (1)

Phát biểu bài toán

Cho dãy có n số X = (X(1), X(2), ..., X(n)).

Hãy tìm dãy con không giảm dài nhất.

Nghĩa là, tìm

$$S = (X(p_1), X(p_2), ..., X(p_k)),$$

thỏa mãn

$$X(p_1) \le X(p_2) \le ... \le X(p_k),$$

với $p_1 < p_2 < ... < p_k$,

sao cho k lớn nhất.

5. Dãy con đơn điệu dài nhất (2)

Ví dụ minh họa:

Cho dãy

CS	1	2	3	4	5	6	7	8	9
X=	**12**	**16**	**5**	**9**	**6**	**4**	**7**	**11**	**5**

thì dãy con không giảm dài nhất có 4 phần tử là {X(3), X(5), X(7), X(8)}

5. Dãy con đơn điệu dài nhất (3)

Trạng thái khởi tạo, quyết định ban đầu

Kí hiệu L(i) là nhãn của X(i) trong dãy không giảm dài nhất mà nó có thể tham gia:

L(1) = 1;

CS	1	2	3	4	5	6	7	8	9
X=	**12**	**16**	**5**	**9**	**6**	**4**	**7**	**11**	**5**
L=	1								

5. Dãy con đơn điệu dài nhất (4)

Chiến lược tối ưu

Xét X(i), với i>1:

Kí hiệu L(i) là nhãn tốt nhất của X(i). Dãy không giảm dài nhất mà X(i) có thể tham gia là đứng sau X(q):

q<i,

X(q) ≤ X(i).

L(q) = max { L(j): j < i, X(j) ≤ X(i) }

L(i) = L(q) + 1

Ví dụ, i = 2 có q=1.

L(2) = L(1) + 1;

CS	1	2	3	4	5	6	7	8	9
X=	**12**	**16**	**5**	**9**	**6**	**4**	**7**	**11**	**5**
L=	1	2							

5. Dãy con đơn điệu dài nhất (5)

Thuật toán

1) L(1) = 1;
2) for(i=2; i≤*n*; i++)

    {   if (tồn tại X(j)≤X(i) với j<i)

            L(i) = max {L(j): X(j) ≤ X(i), j=1,2,..., i-1}+1

        else L(i) = 1;

    }

CS	1	2	3	4	5	6	7	8	9
X=	**12**	**16**	**5**	**9**	**6**	**4**	**7**	**11**	**5**
L=	1	2	1	2	2	1	3	4	2

5. Dãy con đơn điệu dài nhất (6)

CS	1	2	3	4	5	6	7	8	9
X=	**12**	**16**	**5**	**9**	**6**	**4**	**7**	**11**	**5**
L=	1	2	1	2	2	1	3	**4**	2

Xác định dãy

Xác định dãy con không giảm dài nhất:

1) k = L(r) = max { L(i): i=1,2,...,n}

   r = 8,   k = 4,

5. Dãy con đơn điệu dài nhất (7)

CS	1	2	3	4	5	6	7	8	9
X=	**12**	**16**	**5**	**9**	**6**	**4**	**7**	**11**	**5**
L=	1	2	1	2	2	1	3	**4**	2

Xác định dãy r = 8, k = L(8) = 4

2) p(L(r)) := r;
3) for(j=L(r)-1; j≥1; j--)
   p(j) = max{ t: L(t)=j, t<p(j+1), X(t) ≤ X(p(j+1)}
4) output S={ X(p(1)), X(p(2)), …, X(p(k))}

p(4) = 8,

p(3) = max{ t: L(t)= 3, t<p(4), X(t) ≤ X(p(4)}
= max{ t: L(t)= 3, t< 8, X(t) ≤ 11} = 7

p(2) = max{ t: L(t)= 2, t<p(3), X(t) ≤ X(p(3)}
= max{ t: L(t)= 2, t<7, X(t) ≤ 7} = 5

p(1) = max{ t: L(t)= 1, t<p(2), X(t) ≤ X(p(2)}
= max{ t: L(t)= 1, t<5, X(t) ≤ 6} = 3

5. Dãy con đơn điệu dài nhất (8)

CS	1	2	3	4	5	6	7	8	9
X=	**12**	**16**	**5**	**9**	**6**	**4**	**7**	11	**5**
L=	1	2	**1**	2	**2**	1	**3**	4	2

Xác định dãy r = 8, k = L(8) = 4

2) p(L(r)) := r;

3) for(j=L(r)-1; j≥1; j--)

   p(j) = max{ t: L(t)=j, t<p(j+1), X(t) ≤ X(p(j+1)}

4) output S={ X(p(1)), X(p(2)), ..., X(p(k))}

p(4) = 8,

p(3) = max{ t: L(t)= 3, t<p(4), X(t) ≤ X(p(4)}

= max{ t: L(t)= 3, t< 8, X(t) ≤ 11} = 7

p(2) = max{ t: L(t)= 2, t<p(3), X(t) ≤ X(p(3)}

= max{ t: L(t)= 2, t<7, X(t) ≤ 7} = 5

p(1) = max{ t: L(t)= 1, t<p(2), X(t) ≤ X(p(2)}

= max{ t: L(t)= 1, t<5, X(t) ≤ 6} = 3

output: X(3), X(5), X(7), X(8)

5. Dãy con đơn điệu dài nhất (9)

CS	0	1	2	3	4	5	6	7	8	9
X=	**0**	**12**	**16**	**5**	**9**	**6**	**4**	**7**	11	**5**
L=	0	1	2	**1**	2	**2**	1	**3**	4	2

Chỉnh lại thuật toán

1) $X(0) = 0$; $L(0) = 0$;
2) for ($i=1$; $i \le n$; i++)

   $$L(i) = \max\left\{L(j): X(j) \le X(i),\ j=1,2,\dots, i-1\right\} + 1 \qquad (1)$$

3) $k = L(r) = \max\{ L(i): i=1,2,\dots,n\}$
4) $p(L(r)) = r$;
5) for ($j = L(r)-1$; $j \ge 1$; j--)

   $$p(j) = \max\left\{ t:\ L(t)=j,\ t<p(j+1),\ X(t) \le X(p(j+1))\right\}$$

6) output $S=\left\{ p(1), p(2), \dots, p(k)\right\}$

Độ phức tạp thuật toán $O(n^2)$.

5. Dãy con đơn điệu dài nhất (10)

1) $X(0) = 0$; $L(0) = 0$;
2) for ($i=1$; $i \le n$; i++)
   $L(i) = \max\{L(j): X(j) \le X(i), j=1,2,\dots, i-1\} +1$ (1)
3) $k = L(r) = \max\{ L(i): i=1,2,\dots,n\}$
4) $p(L(r)) = r$;
5) for ($j = L(r)-1$; $j \ge 1$; j--)
   $p(j) = \max\{ t: L(t)=j, t<p(j+1), X(t) \le X(p(j+1))\}$
6) output $S=\{ p(1), p(2), \dots, p(k)\}$

Tính đúng đắn của thuật toán

Mệnh đề:

$L(i)$ độ dài dãy con không giảm dài nhất trong $\{X(1), \dots, X(i)\}$ có $X(i)$ đứng cuối

Rõ ràng mệnh đề đúng với $i = 0, 1$.

Kí hiệu $S^{(i)}$ là dãy con không giảm dài nhất trong $\{X(1), \dots, X(i)\}$ có $X(i)$ đứng cuối.

Viết tắt là: $S^{(i)} = LMS(X(1), \dots, X(i))$

5. Dãy con đơn điệu dài nhất (11)

1) $X(0) = 0; L(0) = 0;$
2) for ($i=1$; $i \le n$; i++)
   $L(i) = \max \{L(j): X(j) \le X(i), j=1,2,\ldots, i-1\} +1$ (1)
3) $k = L(r) = \max \{ L(i): i=1,2,\ldots,n\}$
4) $p(L(r)) = r;$
5) for ($j = L(r)-1$; $j \ge 1$; j--)
   $p(j) = \max\{ t: L(t)=j, t<p(j+1), X(t) \le X(p(j+1))\}$
6) output $S=\{ p(1), p(2), \ldots, p(k)\}$

Giả thiết $L(j) = |S^{(j)}|$, $j = 0, .., i-1$:

$(X(0))$,

$(X(0), X(1))$,

.....

$(X(1), X(2), \ldots, X(i-1))$

Kí hiệu:

$L(q) = \max \{L(j): \quad j=0,..,i-1, X(j) \le X(i)\}$ (2)

$L(i) = L(q) + 1;$

Phải chứng minh $L(i) = |S^{(i)}| = |LMS(X(1), \ldots, X(i))|$

5. Dãy con đơn điệu dài nhất (12)

Kí hiệu $M^{(i)} = LMS(X(1), ..., X(i))$, $|M^{(i)}| = \alpha$

Phải CM: $L(i) = \alpha$

<u>Giả sử</u> $\alpha > L(i)$

Kí hiệu X(k) là phần tử đứng ngay sau X(i) trong $M^{(i)}$

$S^{(k)} = LMS(X(1), ..., X(k))$, $|S^{(k)}| = \alpha - 1$, do bớt đi X(i)

$L(k) = |S^{(k)}| = \alpha - 1$ (theo giả thiết quy nạp)

do $X(k) \le X(i) \Rightarrow L(k) < L(i)$

hay $L(k) \le \alpha - 2$

hay $\alpha - 1 = L(k) \le \alpha - 2$

Mâu thuẫn này chứng tỏ giả sử trên bị sai.

Suy ra $L(r) = |LMS(X(1), ..., X(n))|$

5. Dãy con đơn điệu dài nhất (13)

Bài toán 1:

Có n cuộc họp, cuộc họp thứ i bắt đầu vào thời điểm a_i và kết thúc ở thời điểm b_i, $a_i < b_i$, với i=1, 2, ..., n. Hai cuộc họp được bố trí tại cùng một phòng nếu khoảng thời gian làm việc của chúng giao nhau không nhiều hơn thời điểm tiếp nối.

Hãy xếp lịch các cuộc họp để sao cho số phòng phải sử dụng là ít nhất.

5. Dãy con đơn điệu dài nhất (15)

<u>Lời giải gần đúng</u>:

1) Sắp xếp các cuộc họp sao cho $a_1 \le a_2 ... \le a_n$.
2) Định nghĩa cuộc họp j "lớn hơn" cuộc họp i nếu $b_i \le a_j$;
3) Đặt m = n; X = $\{(a_1, b_1), ..., a_n, b_n)\}$; k = 1;
4) while ($X \neq \emptyset$)
    a) Dùng LMS giải bài toán với dãy X;
    b) Lưu dãy kết quả vào mảng P_k;
    c) k ++;
    d) Loại dãy con tìm được ra khỏi X;

5) output: k-1 và các dãy $P_1, P_2, ..., P_{k-1}$;

5. Dãy con đơn điệu dài nhất (16)

Phương pháp vét cạn giải bài toán LMS

1) S* = ∅; L* = 0;
2) for each (dãy con S của X)

   if (S không giảm)

      L = độ dài của S;

      if (L > L*)

      {L = L*; S* = S;}

3) output L*, S*;

Kỹ thuật tìm dãy con của X có thể sử dụng kỹ thuật tìm dãy nhị phân độ dài *n*.

Độ phức tạp của thuật toán $O(2^n)$.

5. Dãy con đơn điệu dài nhất (17)

Phương pháp gần đúng giải bài toán LMS

1) Chọn phần tử bé nhất a_t thuộc X;
2) k = 1; p(k) = t;
3) for (j = t+1; j≤n; j++)

```
    if (X(p(k)) ≤ X(j)) )
        {   k++;
            p(k) = j;
        }
```

4) output: k và p(1), ..., p(k);

Độ phức tạp O(n).

5. Dãy con đơn điệu dài nhất (18)

So sánh lý thuyết các thuật toán vét cạn, gần đúng và quy hoạch động giải bài toán LMS.

Thuật toán vét cạn dựa trên sinh dãy nhị phân độ dài *n*:
$O(2^n)$;
Thuật toán gần đúng:
$O(n)$;
Thuật toán quy hoạch động: $O(n^2)$

So sánh thực nghiệm (bài tập tự làm)

Quy hoạch động

1. **Bài toán ba lô (knapsack 0-1)**
2. **Nguyên lý tối ưu Belman**
3. **Đánh giá thuật toán**
4. **Bài toán dãy con chung dài nhất**
5. **Bài toán dãy con đơn diệu dài nhất**
6. **Bài toán nhân ma trận**
7. **Bài tập**

Matrix Multiplication

Problem

Cho dãy ma trận $A_1, A_2, \ldots, A_n$,

Kích thước ma trận A_i là $d_{i-1} \times d_i$, $i = 1, 2, \ldots, n$

Tính ma trận

$$A_1 \times A_2 \times \ldots \times A_n$$

Số phép nhân ít nhất caagn thực hiện là bao nhiêu?

Matrix Multiplication (2)

Cho	**A**	**kích thước**	**$m \times n$**
	B	**kích thước**	**$n \times p$**

Tính $C = A \times B$

C(i,j) = A(i,1)*B(1,j) + A(I,2)*B(2,j) +…+ A(I,n)*B(n,j)

i = 1,..m, j = 1,..p

Số phép nhân: **$m \times n \times p$**

Matrix Multiplication (2)

Ví dụ 1: $A \times B \times C$

	kích thước:	40×1	1×40	40×10	
A(BC)	40×1	$(1 \times 40$	$40 \times 10)$	chi phí	1.40.10 = 400
	40×1	1×10		chi phí	40.1.10 = 400
				cộng	**800**
(AB)C	$(40 \times 1$	$1 \times 40)$	40×10	chi phí	40.1.40 = 1,600
	40×40		40×10	chi phí	40.40.10 =16,000
				cộng	**17,600**

Matrix Multiplication (3)

Ví dụ 2

$$\mathbf{A} \times \mathbf{B} \times \mathbf{C} \times \mathbf{D}$$

$$40{\times}1 \quad 1{\times}40 \quad 40{\times}10 \quad 10{\times}50$$

Có 5 phương án nhân:

(AB)   (CD)
[(AB)   C]   D
[A (BC)]   D
A   [(BC)   D]
A   [B   (CD)]

Matrix Multiplication (4)

Example 2

A × B × C × D

40×1 1×40 40×10 10×50

(AB) (CD)	M = AB: 40x40 40*40=1,600	P = CD: 40x50 10*50*40=20,000	T= MP: 40x50 40*50*40=80,000	101,600
[(AB) C] D	M=AB: 40×40 40*40=1,600	P= MC: 40×10 40*10*40=16,000	T=PD: 40×50 20*50*40=40,000	57,600
[A (BC)] D	M=BC: 1×10 1*40*10=400	P=AM: 40×10 40*10 = 400	T=PD: 40×50 20*50*40=40,000	40,800
A [(BC) D]	M=BC: 1×10 1*40*10=400	P=MD: 1×50 10*50 = 500	T=AP: 40×50 40*50=2,000	2,900
A [B (CD)]	M=CD: 40×50 10*50*40=20,000	P=BM: 1×50 40*50 = 2,000	T=AP: 40×50 40*50=2000	24,000

Matrix Multiplication (5)

Ký hiệu

$$M_{ij} = A_i \times A_{i+1} \times \ldots \times A_{j-1} \times A_j \text{ , } i \leq j$$

nếu

B[i, j] = số ít nhất phép nhân để tính M_{ij}

thì

B[1,n] - số ít nhất phép nhân để tính $A_1 \times A_2 \times \ldots \times A_n$

Matrix Multiplication (6)

Cách tính:

$$M_{ij} = (A_i \times A_{i+1} \times \ldots \times A_k) \times (A_{k+1} \times \ldots \times A_{j-1} \times A_j)$$

<u>chi phí</u>:

$$B[i, j]_k = B[i,k] + d_{i-1} d_k d_j + B[k+1,j]$$

$$B[i, j] = \min \{ B[i, j]_k \quad : i \le k < j \}$$

$$= \min \{ B[i,k] + d_{i-1} d_k dj + B[k+1,j] \quad : i \le k < j \}$$

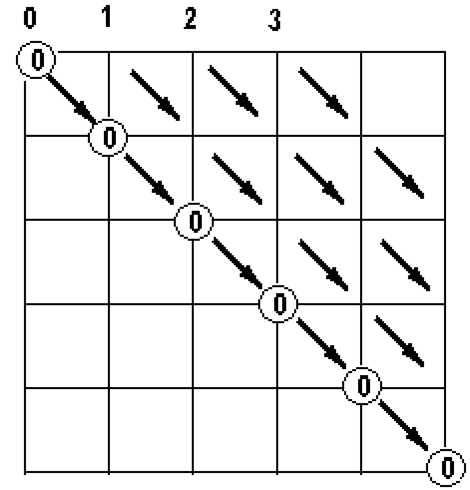
Matrix Multiplication (7)

$$M_{ij} = (A_i \times A_{i+1} \times \ldots \times A_k) \times (A_{k+1} \times \ldots \times A_{j-1} \times A_j)$$

$$B[i, j] = \min \{ B[i,k] + d_{i-1} d_k d_j + B[k+1,j] \}$$

d_0	d_1	d_2	d_3	d_4
40	1	40	10	50

0	1600	800	2900
	0	400	900
		0	20,000
			0



B[i, i] =0, i =1,..,4;

B[1,2] = min {B[1,1] + B[2,2] + $d_0 d_1 d_2$} = 1600

B[2,3] = min {B[2,2] + B[3,3]+ $d_1 d_2 d_3$} = 400

B[3,4] = min {B[3,3] + B[4,4]+ d2d3d4} = 20,000

B[1,3]=min{B[1,1]+B[2,3]+$d_0 d_1 d_3$, B[1,2]+B[3,3]+$d_1 d_2 d_3$=min{800, 2000} = 8000

B[2,4]=min{B[2,2]+B[3,4]+$d_1 d_2 d_4$, B[2,3]+B[4,4]+$d_1 d_3 d_4$=min{22,000, 900} = 900

B[1,4] =min{B[1,1]+B[2,4]+$d_0 d_1 d_4$, B[1,2]+B[3,4]+$d_0 d_2 d_4$, B[1,3]+B[4,4]+$d_0 d_3 d_4$}
=min{2900, 41600,20400} = 2900

Matrix Multiplication (8)

1. for i = 1 to n   do   B[[i,i] = 0;
2. for t = 1 to n-1 do

    for i =1 to n-t do

        B[i,i+t] = min{ B[i,k]+B[k+1,i+t] +$d_{i-1}d_kd_{i+t}$: i ≤ k ≤ i+t -1}

        DC[i,i+t] = chỉ số k mà tại đó tìm được B[i,i+t];

3. output B[1,n];

Complexity: O(n^3)

Matrix Multiplication (9)

```
void        Viết biểu thức  (i, j : Chỉ số);
     if  (i=j)               Viết( ‘A’, i)
     else
             k := DC[i, j];
             Viết(‘(‘);
             Viết biểu thức  (i, k);
             Viết(‘*’);
             Viết biểu thức  (k+1,j );
             Viết(‘)‘);
```

Bài tập

<u>Câu hỏi ôn tập và bài tập tự làm 1</u>:

1. **Có *n* cuộc họp, cuộc họp thứ *i* bắt đầu vào thời điểm *a*i và kết thúc ở thời điểm *b*i. Hai cuộc họp được bố trí tại cùng một phòng nếu khoảng thời gian làm việc của chúng giao nhau không nhiều hơn thời điểm tiếp nối. Hãy bố trí sao cho số phòng phải sử dụng là ít nhất.**
2. **Mỗi đoạn thẳng trên trục O*x* được mô tả bởi hai giá trị [*a*, *b*]. Kí hiệu S là tập hợp *n* đoạn thẳng S = { [a_i, b_i], *i* = 1,2,...,*n*}. Xây dựng thuật toán tìm tập con S* có nhiều phần tử nhất thuộc S sao cho các đoạn thẳng trong S* đôi một không có điểm chung.**

Bài tập

Câu hỏi ôn tập và bài tập tự làm 1:

3. **Mỗi đoạn thẳng trên trục Ox được mô tả bởi hai giá trị [a, b]. Kí hiệu S là tập hợp n đoạn thẳng S = { [a_i, b_i], i = 1,2,...,n}. Xây dựng thuật toán tìm tập S* sao cho các đoạn thẳng trong S* đôi một không có điểm chung mà có tổng độ dài các đoạn thẳng là lớn nhất.**
4. **Mỗi đoạn thẳng trên trục Ox được mô tả bởi hai giá trị [a, b]. Kí hiệu S là tập hợp n đoạn thẳng S = { [a_i, b_i], i = 1,2,...,n}. Xây dựng thuật toán tìm tập S* gồm các đoạn thẳng "liên thông", nghĩa là các đoạn thẳng trong S* khi đặt trên trục hoành sẽ tạo thành một đoạn thẳng mới, sao cho đoạn thẳng mới được tạo thành có độ dài lớn nhất.**