

Algorithm Design and Analysis

วิชาบังคับก่อน: 204251 และ 206281

ผู้สอน: ตอน 1 อ. เบนจามาต ปัญญางาม เรียน ห้อง 207
 ตอน 2 อ. ดร. จักริน ขวชาติ เรียน ห้อง 209

วันสอบปลายภาค : วันศุกร์ที่ 13 พ.ย. 2563
 เวลา 8:00 -11:00 น. (ตามประกาศมหาวิทยาลัย)

บทที่ 9 Dynamic Programming Part II

อ. ดร. จักริน ขวชาติ
 อ. เบนจามาต ปัญญางาม

Comp science CMU

Maximum Value Contiguous Subsequence

อ. ดร. จักริน ขวชาติ
 อ. เบนจามาต ปัญญางาม

Comp science CMU

Maximum Value Contiguous Subsequence

กำหนดให้ ลำดับของเลขจำนวนจริง (มีทั้งเลขบวกและเลขลบ) n ตัว

$$A_1, A_2, \dots, A_n$$

เป้าหมาย หาลำดับที่ติดกัน A_i, \dots, A_j ที่ผลรวมของสมาชิกในลำดับนั้นมีค่ามากที่สุด

ตัวอย่างเช่น

□ กำหนดลำดับของเลขเป็น 1, 3, 5, -4, 2

จะได้ค่าผลรวมมากที่สุดเป็น 9 (เมื่อเลือกลำดับที่ติดกันเป็น 1, 3, 5)

□ กำหนดลำดับของเลขเป็น 1, -5, 2, -1, 3 จะได้ค่าผลรวมมากที่สุดเท่าไร?

อ. ดร. จักริน ขวชาติ
 อ. เบนจามาต ปัญญางาม

Comp science CMU

Solution by Dynamic programming

Input : array $A[1.. n]$ ของเลขจำนวนจริง



$$\text{Goal: } \max \sum_{x=i}^j A[x]$$

1. นิยาม subproblem

ให้ $B[j]$ แทนผลรวมที่ติดกันที่มากที่สุดเมื่อพิจารณาถึงช่องที่ j

อ. ดร. จักริน ขวชาติ
 อ. เบนจามาต ปัญญางาม

Comp science CMU

Solution by Dynamic programming

5

2. หา recurrence ของ subproblem

เนื่องจากสิ่งที่ต้องการคือ หาลำดับที่ติดกัน A_1, \dots, A_j ที่ให้ผลรวมมากที่สุด

ดังนั้นพิจารณาตัวที่ j

คำถาม: เมื่อพิจารณาตัวเลขตัวที่ j สิ่งที่เกิดขึ้นได้มีอะไรบ้าง

คำตอบ: จะตัดสินใจเลือกรวมตัวที่ j หรือไม่รวมตัวที่ j

คำถาม: การเลือกรวมตัวที่ j เพราะเหตุผลอะไร

คำตอบ: เมื่อนำผลรวมนับรวมตัวที่ j แล้วทำให้ผลรวมมีค่ามากขึ้นโดยมากกว่า การเริ่มต้นนับใหม่ที่ตำแหน่งที่ j

คำถาม: การเลือกที่จะไม่รวมตัวที่ j เพราะเหตุผลอะไร

คำตอบ: การเริ่มต้นนับใหม่จะทำให้ได้ผลรวมมากกว่า

Solution by Dynamic programming

6

2. หา recurrence ของ subproblem

ตัวอย่างกำหนดลำดับของเลขเป็น 1, -5, 2, -1, 3 จะได้ค่าผลรวมมากที่สุดเท่าไร?

ค่า j ค่า $A[j]$ ผลรวมลำดับติดกัน

0 1 มีตัวเดียว 1

1 -5 จะเลือกระหว่าง $1 + -5 + ?$ และ $-5 + ?$

คำตอบคือเลือก 1 + -5 เพราะการนับรวมตัวที่ j แล้วทำให้ผลรวมมีค่ามากขึ้น

2 2 จะเลือกระหว่าง $1 + -5 + 2 + ?$ และ $2 + ?$

คำตอบคือเลือก 2 เพราะการนับรวมตัวที่ j แล้วไม่ได้ทำให้ผลรวมมีค่ามากขึ้น

3 -1 จะเลือกระหว่าง $2 + -1 + ?$ และ $-1 + ?$

คำตอบคือเลือก 2 + -1

Solution by Dynamic programming

7

2. หา recurrence ของ subproblem

$$B[j] = \max\{B[j-1] + A[j], A[j]\}$$

โดยที่

$B[j] = B[j-1] + A[j]$ คือ ผลรวมที่ติดกันที่มากที่สุดกรณีที่ย้ายช่วงของคำตอบมารวมช่องที่ j

$B[j] = A[j]$ คือ ผลรวมที่ติดกันที่มากที่สุดกรณีการเริ่มต้นนับใหม่ที่ตำแหน่งที่ j ($A[j]$) เป็นต้นไป

3. หา base case

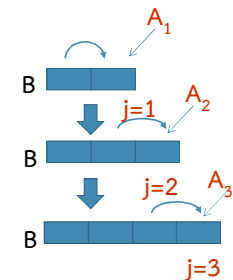
ถ้ามีตัวเดียวก็ต้องตอบเลย ไม่ว่าจะติดลบหรือไม่ก็ตาม

$$B[0] = A[0]$$

Dynamic programming algorithm

8

```
int maxContiguousSum(int A[], int len) {
    int j;
    int B[len];
    B[0] = A[0];
    for (j = 1; j < len; j++) {
        B[j] = max(B[j-1]+A[j], A[j] );
    }
    int max_so_far = B[0];
    for (j = 1; j < len; j++) {
        if (max_so_far < B[j])
            max_so_far = B[j];
    }
    return max_so_far;
}
```



Dynamic programming algorithm(Update version)

```

int maxContiguousSum(int A[], int n)
{
    int j;
    int max_so_far = A[0], i;
    int curr_max = A[0];
    for (j = 1; j < n; j++) {
        curr_max = max(curr_max + A[j], A[j]);
        max_so_far = max(curr_max, max_so_far);
    }
    return max_so_far;
}

```

แบบฝึกหัด

□ -2, 11, -4, 13, -5, 2

□ -15, 29, -36, 3, -22, 11, 19, -5

Longest Increasing Subsequence

Longest Increasing Subsequence

กำหนดให้ ลำดับของตัวเลข A_1, A_2, \dots, A_n

เป้าหมาย ต้องการหา subsequence (ส่วนของลำดับไม่จำเป็นต้องติดกัน) ที่เพิ่มขึ้นที่ยาวที่สุด

ตัวอย่างเช่น

ลำดับของตัวเลขคือ 1, 5, 3, 4, 8, 2, 6, 7

จะได้ subsequence ที่เพิ่มขึ้นที่ยาวที่สุดคือ 5 ตัว

ได้แก่ 1, 3, 4, 6, 7

Solution by Dynamic programming

13

1. นิยาม sub problem

ให้ $L(j)$ ผลการนับ subsequence ที่เพิ่มขึ้นที่ยาวที่สุด ณ ตัวที่ j

Solution by Dynamic programming

14

1. นิยาม sub problem

ให้ $L(j)$ แทน ผลการนับ subsequence ที่เพิ่มขึ้นที่ยาวที่สุด ณ ตัวที่ j

2. หา recurrence ของ subproblem

พิจารณาตัวที่ j

คำถาม ตัวที่ j ควรจะนับต่อจากตัวไหน

คำตอบ ตัวที่มีค่าน้อยกว่ามันและมีผลการนับที่มากที่สุด

Solution by Dynamic programming

15

$L(j)$ แทน ผลการนับ subsequence ที่เพิ่มขึ้นที่ยาวที่สุด ณ ตัวที่ j

ตัวอย่างลำดับของตัวเลขคือ 1, 5, 3, 4, 8, 2, 6, 7

พิจารณาถึงตัวที่ j

j	$A(j)$	$L(j)$	//พิจารณาทุกตัวที่ $A[i] < A(j), i=0 \dots j-1$ แล้วเก็บค่ามากที่สุด
0	1	1	//ตัวแรกนับได้ 1 ตัว
1	5	2	//เลือก max โดยคิดจากกรณี $L(0) + 1 \rightarrow 2$
2	3	2	//เลือก max โดยคิดจากกรณี $L(0) + 1 \rightarrow 2$
3	4	3	//เลือก max โดยคิดจากกรณี $L(0) + 1 \rightarrow 2, L(2) + 1 \rightarrow 3$ เพราะว่า $A[0], A[2] < A(3)$
4	8	4	//เลือก max โดยคิดจากกรณี $L(0) + 1 \rightarrow 2, L(1) + 1 \rightarrow 3,$ $L(2) + 1 \rightarrow 3, L(3) + 1 \rightarrow 4$ เพราะทุกตัว $< A(4)$
5	2	2	// $L(5)$ เลือก max โดยคิดจากกรณี $L(0) + 1 \rightarrow 2$

Solution by Dynamic programming

16

2. หา recurrence ของ subproblem

$$L(j) = \max_{i < j; A[i] < A[j]} \{L(i)\} + 1$$

เมื่อ $\max_{i < j; A[i] < A[j]} \{L(i)\}$ คือ $L(i)$ ที่มากที่สุดที่พิจารณาจากลำดับที่ i ที่น้อยกว่า j ที่ $A[i] < A[j]$

3. หา base case

$$L(1) = 1$$

อย่าลืม เราจะหาค่ามากที่สุด แต่ว่าตอนนี้เราดูที่ว่าถ้านับเอาตัวที่ j แล้วจะทำให้มากที่สุดอย่างไร ไม่อย่างนั้นต้องงวนหาค่ามากที่สุดอีกครั้งด้วย

Dynamic programming algorithm

```
int LIS(A[], n){
    for (i = 1 to n) L[i]=1
    max = L[1]
    for (i = 2 to n) {
        for (j = 1 to i - 1) {
            if (A[i] > A[j] && L[i] < L[j] + 1)
                L[i] = L[j] + 1
            if(max<L[i])
                max=L[i]
        }
    }
    return max
}
```

อ. ดร. จักรีน ชวชาลี
อ. ณัฐธาดา ปิณฑุภากร