เอกสารประกอบการอบรมคอมพิวเตอร์โอลิมปิกวิชาการ ค่าย 2
27 มีนาคม – 22 เมษายน 2563

กำหนดการพลวัต
(Dynamic Programming)

ขั้นตอนวิธีแบบ Dynamic Programming สำหรับปัญหาการหาค่าที่ดีที่สุด

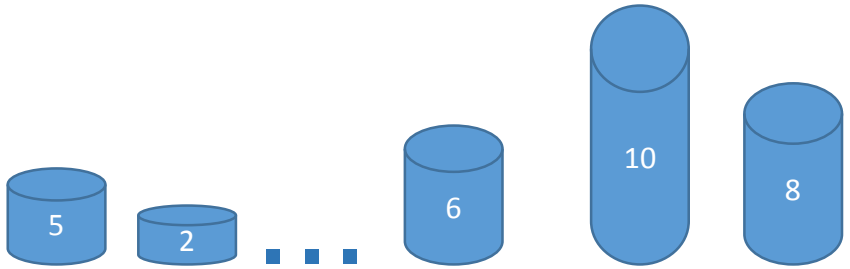
ขั้นตอนการแก้ปัญหาการหาค่าที่ดีที่สุด แบ่งออกเป็น 4 ขั้นตอน

- กำหนดลักษณะของคำตอบที่ดีที่สุดของปัญหาที่มีขนาด n ใดๆ
- นิยามความสัมพันธ์เวียนเกิด (recurrence relation) สำหรับปัญหา
- คำนวณหาค่าของคำตอบที่ดีที่สุด (the value of an optimal solution) โดยใช้วิธีการจากล่างขึ้นบน (Bottom-Up Dynamic Programming)
- หาคำตอบที่ดีที่สุด (optimal solution) จากค่าที่คำนวณได้

OP#1: ปัญหาแถวเหรียญ (Coin-row problem)

ปัญหา : ให้เหรียญ n เหรียญวางเรียงกันแถวตรง โดยเหรียญมีมูลค่าเป็น c[1],c[2],…,c[n] ตามลำดับ จงหาจำนวนเงินรวมที่มากที่สุด จากการหยิบเหรียญโดยที่ไม่มีสองเหรียญใดอยู่ติดกันของแถวเหรียญเริ่มต้น

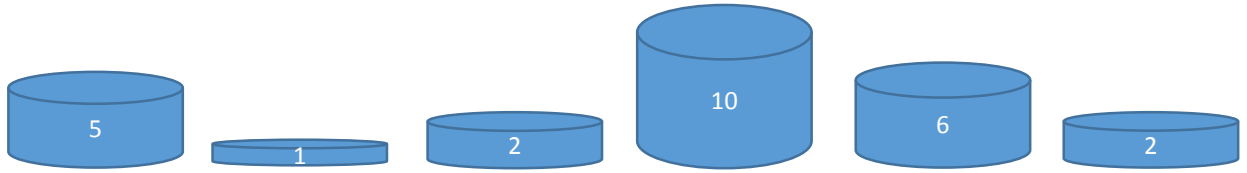
วิธีทำ



เหรียญที่	1	2	…	n-2	n-1	n
มูลค่า	C[1]	C[2]	…	C[n-2]	C[n-1]	C[n]

- ให้ F(n) : จำนวนเงินรวมที่มากที่สุดของแถวเหรียญ n เหรียญ

ปัญหาแถวเหรียญ (Coin-row problem)

ตัวอย่าง 

เหรียญที่ i	1	2	3	4	5	6
มูลค่า C	5	1	2	10	6	2

- ให้ **F(n) :** จำนวนเงินรวมที่มากที่สุดของแถวเหรียญ **n** เหรียญ

i	0	1	2	3	4	5	6
C		5	1	2	10	6	2
F							

F(0)=

i	0	1	2	3	4	5	6
C		5	1	2	10	6	2
F							

F(1)=

i	0	1	2	3	4	5	6
C		5	1	2	10	6	2
F							

F(2)=

ปัญหาแถวเหรียญ (Coin-row problem)

i	0	1	2	3	4	5	6
C		5	1	2	10	6	2
F							

F(3)=

i	0	1	2	3	4	5	6
C		5	1	2	10	6	2
F							

F(4)=

i	0	1	2	3	4	5	6
C		5	1	2	10	6	2
F							

F(5)=

i	0	1	2	3	4	5	6
C		5	1	2	10	6	2
F							

F(6)=

ปัญหาแถวเหรียญ (Coin-row problem)

ปัญหา : ให้เหรียญ n เหรียญวางเรียงกันแถวตรง โดยเหรียญมีมูลค่าเป็น c[1],c[2],...,c[n] ตามลำดับ จงหาจำนวนเงินรวมที่มากที่สุด จากการหยิบเหรียญโดยที่ไม่มีสองเหรียญใดอยู่ติดกันของแถวเหรียญเริ่มต้น

วิธีทำ

เหรียญที่	1	2	...	n-2	n-1	n
มูลค่า	C[1]	C[2]	...	C[n-2]	C[n-1]	C[n]

- ให้ F(n) : จำนวนเงินรวมที่มากที่สุดของแถวเหรียญ n เหรียญ
- แบ่งเป็น 2 กรณี
  - หยิบเหรียญที่ n จะได้ จำนวนเงินรวมที่มากที่สุด =
  - ไม่หยิบเหรียญที่ n จะได้ จำนวนเงินรวมที่มากที่สุด =

จะได้ F(n)= สำหรับ n>= .....

ปัญหาแถวเหรียญ (Coin-row problem)

- เงื่อนไขเริ่มต้น

  F(0)=…… , F(1)= ……

- สรุปความสัมพันธ์เวียนเกิดและเงื่อนไขเริ่มต้นที่สอดคล้องกับปัญหาคือ

  F(n)=max{ } สำหรับ n>= ……

  และ F(0)=…… , F(1)= ……

ปัญหาแถวเหรียญ (Coin-row problem)

- รหัสเทียม

```
coinrow(C[1],C[2],…,C[n])
    F[0]=…….
    F[1]=…….
    for i=2 to n
        F[i]=……..
    endfor
    return F[n]
```

Quiz

- จงเขียนโปรแกรมภาษา C/C++ แก้ปัญหาแถวเหรียญ (Coin-row problem)
- โดยที่
  - ข้อมูลนำเข้า
    - บรรทัดแรก n เป็นจำนวนเหรียญ
    - บรรทัดที่สอง c[1],c[2],...,c[n] เป็นมูลค่าของเหรียญที่ 1,2,...,n
  - ข้อมูลออก
    - บรรทัดแรก เป็น จำนวนเงินรวมที่มากที่สุดของแถวเหรียญ n เหรียญ

ตัวอย่างข้อมูลนำเข้า	ตัวอย่างข้อมูลส่งออก
6 5 1 2 10 6 2	17

แบบฝึกหัด
ปัญหาการแลกเหรียญ (Change-making problem)

ปัญหา : จงหาจำนวนวิธีในการแลกเงิน n บาท เป็นเหรียญ b[1]<b[2]<…<b[m] และ b[1]=1 โดยใช้จำนวนเหรียญที่น้อยที่สุด

วิธีทำ ให้ F(n) : จำนวนเหรียญที่น้อยที่สุด ซึ่งมีมูลค่ารวมเท่ากับ n บาท

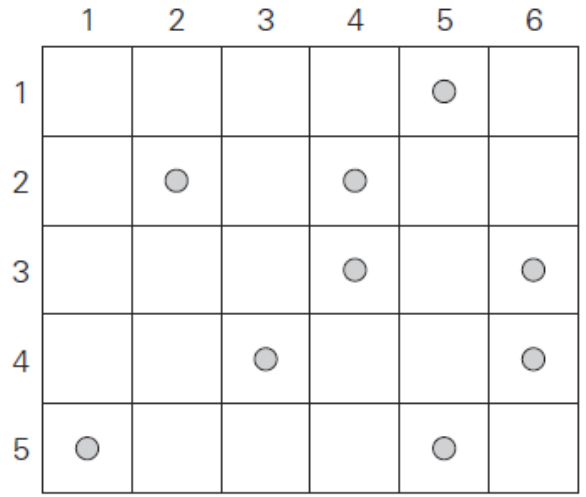
แบบฝึกหัด
ปัญหาการสะสมเหรียญ (Coin-collecting problem)

ปัญหา : ให้เหรียญหลายๆเหรียญวางบนเซลล์ของกระดานขนาด **nxm** โดยเซลล์หนึ่งเซลล์มีเหรียญวางได้ไม่เกินหนึ่งเหรียญ หุ่นยนต์ซึ่งถูกวางอยู่ที่ตำแหน่งเซลล์บนซ้ายของกระดาน ในแต่ละขั้นตอนของการเคลื่อนที่ของหุ่นยนต์ หุ่นยนต์สามารถเคลื่อนที่ไปเซลล์ทางขวาหรือไปเซลล์ด้านล่างหนึ่งเซลล์ของเซลล์ปัจจุบัน เมื่อใดก็ตามที่หุ่นยนต์เคลื่อนที่ไปเซลล์ที่มีเหรียญ หุ่นยนต์จะต้องเก็บเหรียญเสมอ จงหาจำนวนเหรียญที่มากที่สุดที่หุ่นยนต์สามารถเก็บได้

วิธีทำ ให้ **F(i,j) :** จำนวนเหรียญที่มากที่สุดที่หุ่นยนต์สามารถเก็บได้ ณ เซลล์ตำแหน่ง **(i,j)**

c(i,j) = 1 เมื่อ ตำแหน่ง **(i,j)** มีเหรียญ

และ **c(i,j) = 0** เมื่อ ตำแหน่ง **(i,j)** ไม่มีเหรียญ



แบบฝึกหัด
ปัญหาการสะสมเหรียญ (Coin-collecting problem)

ตัวอย่าง