

1. (10 คะแนน) จงเขียนประสิทธิภาพเชิงเวลาของความสัมพันธ์เวียนเกิดต่อไปนี้ โดยใช้ Master Theorem หรือตอบว่า “ไม่ตรงเงื่อนไข” กรณีที่ความสัมพันธ์ดังกล่าวไม่ตรงกับเงื่อนไขของ Master Theorem สำหรับทุก ๆ ข้อให้ถือว่า $T(1) = \theta(1)$

Master Theorem

The Master Theorem applies to recurrences of the following form:

$$T(n) = aT(n/b) + f(n)$$

where $a \geq 1$ and $b > 1$ are constants and $f(n)$ is an asymptotically positive function.

There are 3 cases:

1. If $f(n) = O(n^{\log_b a - \epsilon})$ for some constant $\epsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$.
2. If $f(n) = \Theta(n^{\log_b a} \log^k n)$ with $k \geq 0$, then $T(n) = \Theta(n^{\log_b a} \log^{k+1} n)$.
3. If $f(n) = \Omega(n^{\log_b a + \epsilon})$ with $\epsilon > 0$, and $f(n)$ satisfies the regularity condition, then $T(n) = \Theta(f(n))$.
Regularity condition: $af(n/b) \leq cf(n)$ for some constant $c < 1$ and all sufficiently large n .

ข้อ	Recurrent	ความซับซ้อนเชิงเวลา
(1)	$T(n) = 2^n T(n/2) + n^n$	$\theta(n^{\log_2 n})$
(2)	$T(n) = 2T(n/2) + n \log n$	$\theta(n \log n)$
(3)	$T(n) = 2T(n/4) + n^{0.51}$	$\theta(n^{0.51})$
(4)	$T(n) = 16T(n/4) + n!$	$\theta(n!)$
(5)	$T(n) = \sqrt{2}T(n/2) + \log n$	$\theta(\sqrt{n})$
(6)	$T(n) = 2T(n/2) + n/\log n$	$\theta(n)$
(7)	$T(n) = 64T(n/8) - n^2 \log n$	\times
(8)	$T(n) = 4T(n/2) + \log n$	$\theta(n^2)$
(9)	$T(n) = T(n/2) + n(2 - \cos(n))$	$\theta(n \sqrt{2 - \cos(n)})$
(10)	$T(n) = 4T(n/2) + cn$	$\theta(n^2)$

2. (3 คะแนน) กำหนดให้ผลลัพธ์ของการคำนวณปัญหา Longest Common Subsequence (LCS) ของสายอักขระ OLAHATLASOA กับ AHAHOLALASA ด้วยวิธี Dynamic Programming ตามเอกสารประกอบคำสอน มีผลเป็นดังตารางดังต่อไปนี้จงหาว่า LCS ของสายอักขระสองอันนี้คืออะไร

		O	L	A	H	A	T	L	A	S	O	A
		0	0	0	0	0	0	0	0	0	0	0
A		0	0	0	1	1	1	1	1	1	1	1
H		0	0	1	1	2	2	2	2	2	2	2
A		0	0	0	1	2	3	3	3	3	3	3
H		0	0	0	1	2	3	3	3	3	3	3
O		0	1	1	1	2	3	3	3	3	3	4
L		0	1	2	2	2	3	3	4	4	4	4
A		0	1	2	3	3	3	4	5	5	5	5
L		0	1	2	3	3	3	4	5	5	5	5
A		0	1	2	3	3	4	4	4	5	5	6
S		0	1	2	3	3	4	4	4	5	6	6
A		0	1	2	3	3	4	4	4	5	6	7

คำตอบ _____ (ให้ตอบ string ที่เป็น LCS)

3. (5 คะแนน) Peak Detection ให้ $x[1], x[2], \dots, x[n]$ เป็น รายการของตัวเลขจำนวนเต็มที่มีคุณสมบัติพิเศษว่าตัวเลขในรายการจะเพิ่มขึ้นเรื่อยๆจนถึงตัวเลขที่มีค่ามากที่สุดหลังจากนั้นตัวเลขก็จะลดลงเรื่อยๆจนจบรายการ จงเติม Pseudo Code ที่ใช้หาว่าตัวเลขที่มีค่ามากที่สุดคือค่าอะไร (ตัวอย่างเช่น $x = \langle 1, 2, 5, 6, 8, 9, 15, 3, 2, 1 \rangle$ จะได้ peak คือ 15) โดยประสิทธิภาพเชิงเวลาต้องไม่ช้ากว่า $O(\log n)$ โดยเรารับประกันว่าตัวเลขที่มีค่ามากที่สุดจะไม่เป็นตัวที่ 1 หรือ n และ $n \geq 3$

```

Peak(x, left, right) {
    if (right - left == 2) return x[left+1];
    mid = (left + right) / 2
    if (  $x[mid] > x[mid-1]$  and  $x[mid] > x[mid+1]$  ) return x[mid]
    if (  $x[mid] < x[mid-1]$  ) return Peak(x, left, mid) else return Peak(x, mid, right)
}

Peak(x) {
    return Peak(x, 1, n)
}
    
```

4. (5 คะแนน) สำหรับความสัมพันธ์เวียนเกิด $K(P)$ ที่กำหนดให้ในแต่ละข้อย่อยต่อไปนี้ จงเติมค่าของ $K(P)$ ต่าง ๆ เมื่อคำนวณด้วยวิธี Dynamic Programming ลงในตาราง สำหรับค่า P ตั้งแต่ 0 ถึง 11 และในแต่ละข้อ กำหนดให้ $w_1 = 1, w_2 = 3, w_3 = 5$

$$4.1 \ K(P) = \begin{cases} 0 & P = 0 \\ \min(K(P - w_i) + 1) & (for\ w_i \leq p) \quad P > 0 \end{cases}$$

P	0	1	2	3	4	5	6	7	8	9	10	11
K(P)	0	<u>1</u>	<u>2</u>	<u>1</u>	<u>2</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>2</u>	<u>3</u>	<u>2</u>	<u>3</u>

$$4.2 \ K(P) = \begin{cases} 1 & P = 0 \\ \sum(K(P - w_i)) & (for\ w_i \leq p) \quad P > 0 \end{cases}$$

$$4.2 \ K(P) = \begin{cases} 1 & P = 0 \\ \text{sum}(K(P - w_i)) & \text{(for } w_i \leq p) \end{cases} \quad P > 0$$

on _____

P	0	1	2	3	4	5	6	7	8	9	10	11
K(P)	1	1	1	2	3	5	8	12	17	20	27	34

5. (7 คะแนน) เหมืองแร่แห่งหนึ่งมีพื้นที่เป็นตารางสี่เหลี่ยมจัตุรัส ขนาด 500 คูณ 500 ช่อง ช่องแต่ละช่องสามารถระบุด้วยพิกัด (r, c) ซึ่งหมายถึงช่องในแถวที่ r และคอลัมน์ที่ c ในพื้นที่ของเราสำหรับค่า r, c ตั้งแต่ 0 ถึง 499 เรามีรถขุดแร่อยู่คันหนึ่ง เริ่มต้นที่ตำแหน่ง (499, 499) กำหนดให้เมื่อรถขุดแร่อยู่ในช่อง (a, b) ใด ๆ จะสามารถขุดแร่ได้เป็นจำนวน Mine[a][b] หน่วย หลังจากนั้นรถขุดแร่จะต้องย้ายไปขุดที่อื่น รถขุดแร่นั้นเมื่อขุดช่อง (a, b) เสร็จแล้ว จะสามารถขยับไปยังช่อง (a - 1, b) หรือ (a, b - 1) ช่องใดช่องหนึ่งเท่านั้น แล้วกระทำการขุดแร่ในช่องนั้นต่อไป โดยที่รถขุดแร่ไม่สามารถออกไปนอกพื้นที่เหมืองแร่ได้เด็ดขาด (ตัวอย่างการเดินทางของรถขุดแร่แบบหนึ่งที่เป็นไปได้คือ เริ่มที่ (499,499) -> (499,498) -> (499,497) -> ... -> (499,1) -> (499,0) -> (498,0) -> (497,0) -> ... -> (0,0))

5.1 จงเขียนความสัมพันธ์เวียนเกิด MaxMiner(r, c) ซึ่งคือผลรวมจำนวนแร่ที่ขุดได้มากที่สุดที่เป็นไปได้ เมื่อรถขุดแร่เริ่มจากช่อง (499,499) และเดินทางมาถึงช่อง (r, c) และขุดแร่ในช่อง (r, c) เสร็จเรียบร้อยแล้ว (อย่าลืมเขียนกรณีพื้นฐานด้วย)

$$\text{MaxMiner}(r, c) = \begin{cases} \text{Mine}[499][499] & ; r = 499 \text{ and } c = 499 \\ \max(\text{Miner}(r+1, c) + \text{Mine}[r][c]) & ; r+1 \leq 499 \\ \max(\text{Miner}(r, c+1) + \text{Mine}[r][c]) & ; c+1 \leq 499 \\ \max(\text{MaxMiner}(r+1, c), \text{MaxMiner}(r, c+1)) + \text{Mine}[r][c] & ; r+1 \leq 499 \text{ and } c+1 \leq 499 \end{cases}$$

5.2 จงเขียน pseudo-code ของฟังก์ชัน MaxMiner1(mine, r, c) ตามคำตอบในข้อ 5.1 โดยใช้วิธีการ Divide & Conquer โดยที่ยังไม่ต้องใช้วิธีการ Dynamic Programming โดยฟังก์ชันนี้จะต้องคืนค่า ผลรวมจำนวนแร่ที่ขุดได้มากที่สุดที่เป็นไปได้ เมื่อรถขุดแร่เดินทางมาถึงช่อง (r, c)

```
def MaxMiner1(r, c)
    if (r == 499 and c == 499)
        return Mine[499][499]
    else if (c <= 499 and c+1 > 499)
        return MaxMiner1(r+1, c) + Mine[r][c]
    else if (c+1 <= 499 and r+1 > 499)
        return MaxMiner1(r, c+1) + Mine[r][c]
    else if (c+1 <= 499 and r+1 <= 499)
        return max(MaxMiner1(r, c+1), MaxMiner1(r+1, c)) + Mine[r][c];
```

5.3 จงปรับคำตอบที่ได้ในข้อ 5.2 ให้ใช้วิธีการ Dynamic Programming แบบ Top-Down (หรือที่เรียกว่า Memoization) พร้อมทั้งระบุ ชื่อตัวแปรที่ใช้ในการเก็บค่าของ Dynamic Programming ด้วย โดยให้เขียนเป็น pseudo-code ของฟังก์ชัน MaxMiner2(mine, r, c)

6. (3 คะแนน) ให้พิจารณาปัญหาดาวเด่น (Celebrity Problem) ตามเอกสารประกอบคำสอน กำหนดให้ในกลุ่มมีคนอยู่ 5 คน แต่ละคนกำหนดด้วยหมายเลข 1 ถึง 5 และกำหนดให้มีตาราง a โดยที่ $a[i][j]$ จะมีค่าเป็น 1 ก็ต่อเมื่อคนหมายเลข i รู้จักคนหมายเลข j เป็นดังตารางด้านล่างนี้ ให้ $Celeb(a, p, q)$ เป็นฟังก์ชันที่จะคืนค่าหมายเลขคนที่เป็นดาวเด่น เมื่อเราพิจารณาเฉพาะคนหมายเลข p ถึงหมายเลข q โดยที่ $Celeb(a, p, q)$ จะคืนค่า 0 ก็ต่อเมื่อไม่มีดาวเด่นในกลุ่มคนหมายเลข p ถึง q

เราต้องการคำนวณค่า $Celeb(a, 1, 5)$ ตามวิธี Divide & Conquer ตามเอกสารประกอบคำสอน จงเขียนการเรียกฟังก์ชัน $Celeb$ ที่ถูกเรียกทั้งหมดจากการเรียก $Celeb(a, 1, 5)$ ตามลำดับเวลา พร้อมด้วย Parameter ต่าง ๆ และให้ระบุค่าที่ฟังก์ชันดังกล่าวคืนค่ากลับมาด้วย

ลำดับที่เรียก	ฟังก์ชันและ Parameter	ผลลัพธ์
1.	$Celeb(a, 1, 5)$	
2.		
3.		
4.		
5.		

#	1	2	3	4	5
1	0	1	1	0	0
2	0	0	0	0	0
3	0	1	0	1	1
4	1	0	1	0	1
5	1	1	0	0	0

7. (5 คะแนน) รหัสเทียมต่อไปนี้เป็น การแก้ปัญหาหนึ่งด้วยวิธีการ Dynamic Programming แบบ Bottom-up โดยกำหนดให้มีข้อมูลนำเข้าเป็น Array $A[1..n]$ ส่วนผลลัพธ์ของการคำนวณอยู่ใน Array $B[0..n]$ โดยมีคำตอบอยู่ที่ $B[n]$ จงเขียนความสัมพันธ์เวียนเกิดของฟังก์ชัน $B(x)$ ของรหัสเทียมดังกล่าว พร้อมทั้งระบุกรณีพื้นฐานด้วย

```

B = Array [0..n]
B[0] = 0
for i = 1 to n
    max = 0
    for j = 1 to i-1
        if (a[i] > a[j] and b[j] > max) max = b[j]
    b[i] = max + 1
    
```

$$B(x) = \begin{cases} 0 & ; x = 0 \\ \max(B(t)) + 1 & ; t \in Q \text{ for a set of integers which is in } [1, x] \text{ and } a_t < a_x \end{cases}$$