

2. (8 คะแนน) จงวิเคราะห์เวลาการทำงานของอัลกอริทึมข้างล่างนี้ (ในรูปของ Θ ของฟังก์ชัน n , แสดงวิธีทำด้วย)

<pre> a1(n) { s = 0 for (i=1; i<=n; i++) { for (j=1; j<=n; j*=2) { s = i+j } } return s } </pre>	<p>a1(n) ใช้เวลา เป็น $\Theta(?)$</p>
<pre> a2(n) { a20(n, 0) } a20(n, k) { if (k >= n) return a20(n, k+1) a20(n, k+1) } </pre>	<p>a2(n) ใช้เวลา เป็น $\Theta(?)$</p>
<pre> a3(n) { x = random() // ใช้เวลาคงตัว if (n <= 0) return x return 2 * a30(n-1) } </pre>	<p>a3(n) ใช้เวลา เป็น $\Theta(?)$</p>
<pre> a4(x[1..n], y[1..n]) { mergeSort(x) mergeSort(y) LCS(x, y) } </pre>	<p>a4(x[1..n], y[1..n]) ใช้เวลา เป็น $\Theta(?)$</p>

3. (3 คะแนน) ให้ A เป็นเมทริกซ์ของจำนวนจริงขนาด $n \times n$ ปัญหาที่สนใจคือ ช่วยหาจำนวนสองจำนวนที่ติดกันตามแนวอนใน A ที่มีผลต่างน้อยสุด สมชายตอบทันทีว่า เราก้ใช้วิธีแบ่ง A เป็นสองส่วน ซ้ายกับขวา ไปหาคู่ติดกันที่มีผลต่างน้อยสุดฝั่งซ้าย ไปหาคู่ติดกันที่มีผลต่างน้อยสุดฝั่งขวา แล้วก็หาคู่ติดกันข้ามฝั่งที่มีผลต่างน้อยสุด นำสามคู่นี้มาเปรียบเทียบก็ย่อมได้คำตอบ อยากรทราบว่ วิธีของสมชายใช้เวลาการทำงานเท่าใด

.....

.....

4. (8 คะแนน) Stirling number of the second kind เขียนแทนด้วย $\left\{ \begin{matrix} n \\ k \end{matrix} \right\}$ คือจำนวนวิธีแบ่งข้อมูล n ตัว ออกเป็น k ส่วน (แต่ละส่วนมีข้อมูลอย่างน้อยหนึ่งตัว) บรรยายได้ด้วย recurrence ข้างล่างนี้

$$\left\{ \begin{matrix} n \\ k \end{matrix} \right\} = k \left\{ \begin{matrix} n-1 \\ k \end{matrix} \right\} + \left\{ \begin{matrix} n-1 \\ k-1 \end{matrix} \right\} \text{ for } k > 0 \quad \left\{ \begin{matrix} 0 \\ 0 \end{matrix} \right\} = 1, \text{ และ } \left\{ \begin{matrix} n \\ 0 \end{matrix} \right\} = \left\{ \begin{matrix} 0 \\ n \end{matrix} \right\} = 0$$

- จงแสดงด้วยตัวอย่างให้เห็นว่า การคำนวณจำนวนสเตอร์ลิงจากสูตรข้างบนนี้ มี overlapping subproblems
- จงเขียน pseudocode (หรือจะเขียนเป็นฟังก์ชันภาษาซี หรือเมมที่ถอดภาษาจาวาก็ได้) เพื่อคำนวณหา $\left\{ \begin{matrix} n \\ k \end{matrix} \right\}$
 - แบบ (top-down) recursive + memoization
 - แบบ (bottom-up) dynamic programming

7. (15 คะแนน) ปัญหาการแบ่งเท่า, ให้ตัวเลขจำนวนเต็มบวก $a[1], \dots, a[N]$ โดยที่แต่ละตัวอยู่ระหว่าง 1 ถึง Q ($1 \leq a[i] \leq Q$) โดยเราอยากหาวิธีที่จะแบ่งตัวเลขเหล่านี้ออกเป็นสองกลุ่ม โดยให้ S_1 เป็นผลรวมของ a ที่อยู่ในกลุ่ม 1 และ S_2 เป็นผลรวมของ a ที่อยู่ในกลุ่ม 2 เราอยากทราบว่าเราจะแบ่งกลุ่มอย่างไรให้ $|S_1 - S_2|$ น้อยที่สุดเท่าที่จะเป็นไปได้ ปัญหาข้อนี้ สามารถแก้ได้ด้วยวิธี Dynamic Programming จงออกแบบฟังก์ชัน recurrence และเขียน recurrence ดังกล่าว พร้อมทั้งแสดงให้เห็นว่ามีปัญหาย่อยที่ซ้ำกัน และให้วิเคราะห์ประสิทธิภาพของอัลกอริทึมที่สร้างขึ้นตาม recurrence ดังกล่าว (hint: อัลกอริทึมของคุณควรจะใช้เวลา และ หน่วยความจำเป็น polynomial ของ N และ Q และอย่าลืม initial condition ของ recurrence)
8. (15 คะแนน) มีเขาวงกตแห่งหนึ่งประกอบด้วยห้องเล็ก ๆ ต่อกันเป็นตารางจำนวน $N * M$ ห้องกำหนดให้ห้องแต่ละห้องมีพิกัดกำกับเป็น (x,y) โดยที่ห้องมุมบนซ้ายคือห้องหมายเลข $(0,0)$ และห้องมุมล่างขวาคือห้องหมายเลข $(N-1,M-1)$ เราสามารถเดินทางจากห้องหนึ่งไปยังอีกห้องหนึ่งในเขาวงกตนี้ได้ โดยมีกฎคือถ้าเราอยู่ที่ห้อง (X,Y) ใด ๆ เราจะสามารถเดินทางไปยังห้องที่อยู่ติดกันคือห้อง $(X+1,Y)$ หรือห้อง $(X,Y+1)$ ได้เท่านั้น อย่างไรก็ตาม มีห้องบางห้องที่ "ห้ามเข้า" คือไม่สามารถเดินเข้าไปในห้องนั้นได้เลย กำหนดให้ $A[x][y]$ เป็นตัวแปรที่บอกว่าห้องใดห้ามเข้าบ้าง โดยที่ $A[x][y]$ จะมีค่าเป็น 1 ก็ต่อเมื่อห้องดังกล่าวห้ามเข้า สมมติให้เราอยู่ที่ห้อง $(0,0)$ และต้องการเดินทางไปยังห้อง $(N-1,M-1)$ อยากทราบว่ามียุทธศาสตร์การเดินทางที่**แตกต่างกัน**ทั้งหมดกี่แบบ (วิธีการเดินจะแตกต่างกันถ้ามีเดินผ่านห้องอย่างน้อยหนึ่งห้องที่ไม่เหมือนกัน) จงเขียน recurrence ของฟังก์ชัน $C(x,y)$ ที่ใช้ในการคำนวณว่าถ้าเราอยู่ที่ห้องหมายเลข (x,y) และต้องการเดินทางไปยังห้อง $(N-1,M-1)$ แล้วจะมีวิธีการเดินที่**แตกต่างกัน**ทั้งหมดกี่แบบ และให้ตอบว่าการนำเอา recurrence นี้ไปใช้เขียนโปรแกรมจะทำให้ได้โปรแกรมที่มีประสิทธิภาพในการทำงานเป็นอย่างไร (hint: ข้อนี้ไม่จำเป็นต้องออกแบบอัลกอริทึม และอย่าลืม initial condition ของ recurrence)
9. (15 คะแนน) เราต้องการขับรถจากเชียงใหม่ไปสุโขทัย-โลก แต่เนื่องจากระยะทางไกลมากเราจึงต้องแวะพักข้างทางอยู่เรื่อย ๆ สมมติให้มีจุดแวะพักทั้งหมด N จุด (แต่ละจุดมีหมายเลขกำกับ โดยจุดแรกคือหมายเลข 1 และให้ถือว่าเชียงใหม่และสุโขทัย-โลกเป็นจุดพักหมายเลข 0 และหมายเลข $N+1$) การแวะพักที่จุดหมายเลข i นั้นจะทำให้เรามีแรงที่จะขับไปได้ถึง $r[i]$ จุดพัก (แรงขับตอนนี้ไม่มีการบวกทดกัน) ตัวอย่างเช่น แวะพักที่จุด 5 โดยที่ $r[5]$ มีค่า 4 เราจะสามารถขับไปถึงจุด 6, 7, 8, 9 ได้ แต่ไม่สามารถไปถึงจุด 10 ได้ ต้องแวะพักที่ใดที่หนึ่งระหว่างจุด 6 ถึง 9 ก่อน และถ้าหลังจากนั้นเราเลือกพักที่จุด 6 ต่อจากจุด 5 โดยที่ $r[6]$ มีค่าเป็น 2 นั้น จะทำให้เราสามารถขับไปถึงจุดที่ 7 และ 8 ได้เท่านั้น ไม่สามารถไปถึงจุด 9 ได้ เพราะถือว่าเราพักที่จุด 6 ทำให้เรามีแรงวิ่งไปได้แค่ 2 จุดพักเท่านั้น

การแวะพักจุด i นั้นจะต้องเสียเงินเป็นจำนวน $p[i]$ บาท กำหนดให้ $p[0] = 0$ และให้คิดว่าเราแวะพักที่เชียงใหม่เรียบร้อยแล้วออกเดินทาง กำหนดให้ $0 < r[i] \leq N+1$ จงออกแบบอัลกอริทึมเพื่อหาว่าเราจะต้องเสียเงินน้อยสุดเท่าไรเพื่อไปให้ถึงสุโขทัย-โลก พร้อมทั้งวิเคราะห์ประสิทธิภาพในการทำงาน (hint: ไม่มี hint ครับ ขอให้โชคดี)