FACULTY OF ENGINEERING

CHULALONGKORN UNIVERSITY

2110327 Algorithm Design

YEAR III, First Semester, Mid-term Examination, July 25, 2012, Time 8:30 – 11:30

ชื่อ-นามสกุ	ล เลขประจำตัว 2 1 CR58
<u>หมายเหตุ</u>	
1	. ข้อสอบมีทั้งหมด 9 ข้อในกระดาษคำถามคำตอบ รวม จำนวน 4 หน้า 💮 คะแนนเต็ม 103 คะแนน
2	. ไม่อนุญาตให้นำตำราและเครื่องคำนวณต่างๆ ใดๆ เข้าห้องสอบ
3	
4	ા શાંધા મુખ્ય
5	ป
-	v
7	
8	. ผูทบฏบทเขาชายทุจงทเนการลอบ ตามบระกาศคณะรากรกรรมการตั้ง มีโทษ คือ ได้รับ สัญลักษณ์ F ในรายวิชาที่ทุจริต และพักการศึกษาอย่างน้อย 1 ภาคการศึกษา
	2010 110 5/150 516gs11 05k 1 5ks 10 50 1111 (55/1 58514/1111 1 511110 100 1 1660 1 51 1111 1 511110 1
	รับทราบ
	ลงชื่อนิสิต ()
1. 2. 3.	ข้อใดที่ให้ออกแบบอัลกอริทีมนั้น นิสิตสามารถตอบโดยเขียนบรรยายแนวคิดที่ implement ได้ในทางปฏิบัติ หรือจะเขียนเป็นรหัสเทียม ประกอบแนวคิดที่นำเสนอด้วยก็ได้ และต้องวิเคราะห์ประสิทธิภาพเชิงเวลาของอัลกอริทึมที่นำเสนอด้วย ต้องแสดงวิธีทำทุกข้อ การเขียนคำตอบเพียงอย่างเดียวจะไม่มีคะแนนให้ (ยกเว้นว่าจะเขียนในคำสั่ง) ข้อ 1 – 4 ให้ทำในกระดาษคำถาม ข้อ 5 – 9 ให้ทำในสมุดคำตอบ โดยข้อที่ k ให้ทำในหน้า (k-5)*2 + 1 ถึงหน้า (k-5)*2 + 2
1. (9 ค	ะแนน) จงเติมคำตอบที่ถูกต้องในช่องว่างขีดเส้นใต้ (ไม่ต้องอธิบาย)
	$5^1 + 5^2 + 5^3 + \dots + 5^n = \Theta(\underline{\hspace{1cm}})$
1.2.	ให้ $\mathbf s$ คือสตริงยาว n คำสั่ง $\mathbf s.indexOf("x")$ ในจาวา คืน index ของสตริง $\mathbf s$ ที่มีตัวอักษร $\mathbf x$ ถ้าไม่มี $\mathbf x$ จะคืน -1 คำสั่งนี้ทำงานในเวลา $\mathbf O($)
4.0	
	การหาตัวน้อยสุดของข้อมูลจำนวน n ตัวที่เก็บในอาเรย์ต้องใช้เวลา $\Omega(n)$ ถูกหรือผิด :
	ให้ $T(n) = T(n/10) + 10$ จะได้ว่า $T(n) = \Theta($)
	ให้ $T(n) = T(n-2) + n$ จะได้ว่า $T(n) = \Theta($
1.6.	การเรียงลำดับข้อมูลจำนวนหนึ่งล้านตัวด้วย quicksort ใช้เวลา $\Theta($
1.7.	เมท็อด Arrays.sort(Object[] x) ของจาวารุ่น 6 ใช้ mergesort ในการเรียงข้อมูล แต่เปลี่ยนให้ใช้ insertion sort
	เมื่อช่วงที่ต้องการเรียงมีปริมาณข้อมูลไม่เกิน 7 ตัว การทำเช่นนี้ทำให้ Arrays.sort ใช้เวลา $\Theta($)
1.8.	2 ³⁰⁰ mod 11 มีค่าเท่ากับ
	recurrence ของความยาวของ longest common subsequence คือ

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

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

3. (3 คะแนน) ให้ A เป็นเมทริกซ์ของจำนวนจริงขนาด $n \times n$ ปัญหาที่สนใจคือ ช่วยหาจำนวนสองจำนวนที่ติดกันตาม<u>แนวนอน</u>ใน A ที่มีผลต่างน้อยสุด สมชายตอบทันทีว่า เราก็ใช้วิธีแบ่ง A เป็นสองส่วน ซ้ายกับขวา ไปหาคู่ติดกันที่มีผลต่างน้อยสุดฝั่งซ้าย ไป หาคู่ติดกันที่มีผลต่างน้อยสุดฝั่งขวา แล้วก็หาคู่ติดกันข้ามฝั่งที่มีผลต่างน้อยสุด นำสามคู่นี้มาเปรียบเทียบก็ย่อมได้คำตอบ อยาก ทราบว่า วิธีของสมชายใช้เวลาการทำงานเท่าใด

4. (8 คะแนน) Stirling number of the second kind เขียนแทนด้วย $n \choose k$ คือจำนวนวิธีแบ่งข้อมูล n ตัว ออกเป็น k ส่วน (แต่ ละส่วนมีข้อมูลอย่างน้อยหนึ่งตัว) บรรยายได้ด้วย recurrence ข้างล่างนี้

$${n \brace k} = k \begin{Bmatrix} n-1 \cr k \end{Bmatrix} + {n-1 \brace k-1}$$
 for $k > 0$ ${0 \brace 0} = 1$, ਘਰਿਝ ${n \brace 0} = {0 \brack n} = 0$

- จงแสดงด้วยตัวอย่างให้เห็นว่า การคำนวณจำนวนสเตอริงจากสูตรข้างบนนี้ มี overlapping subproblems
- ullet จงเขียน pseudocode (หรือจะเขียนเป็นฟังก์ชันภาษาซี หรือเมท็อดภาษาจาวาก็ได้) เพื่อคำนวณหา $inom{n}{k}$
 - แบบ (top-down) recursive + memoization
 - แบบ (bottom-up) dynamic programming

5.	(15 คะแนน) ปัญหา Point domination, ให้ p[1], , p[N] เป็น array ของจุดใน 2 มิติ N จุด โดยที่พิกัดของจุด i คือ (p[i].x,
	p[i].y) เราจะบอกว่าจุด i อยู่เหนือ จุด j (หรือ จุด j อยู่ใต้จุด i) ก็ต่อเมื่อ $p[i].x>p[j].x$ และ $p[i].y>p[j].y$ โดยในข้อนี้ ให้ $q[i]$
	์ คือจำนวนจุดที่อยู่ใต้จุด i จงอธิบายอัลกอริทึมที่ใช้หา q[1],, q[N] พร้อมทั้งวิเคราะห์ประสิทธิภาพของอัลกอริทึมที่คุณใช้ โดย
	คุณจะได้คะแนนสูงสุด (หากถูกต้อง) ดังนี้
	5 คะแนน หากอัลกอริทึมคุณใช้เวลา O(N ²)
	10 คะแนน หากอัลกอริทึมคุณใช้เวลา O(N log ² N)
	15 คะแนน หากอัลกอริทึมคุณใช้เวลา O(N log N)
	(hint: คุณสามารถ assume ได้ว่า N = 2M สำหรับบางค่า M ที่เป็นจำนวนเต็ม และ ไม่มีจุดใดๆ ที่มีพิกัด x หรือ y ซ้ำกันเลย
	(p[i].x != p[j].x และ p[i].y != p[j].y สำหรับทุกๆ i และ j), คุณอาจพบว่าการเรียงลำดับข้อมูลช่วยคุณได้)

6. (15 คะแนน) หอคอยคู่, ในเมืองๆหนึ่งมีหอคอยอยู่สองหอ โดยหอคอยแต่ละอันเกิดจากการที่แผ่นหินทรงกระบอกมาซ้อนๆกัน โดยทุกๆทรงกระบอกมีความสูงเท่ากันหมด แต่ว่ารัศมีอาจไม่เท่ากันโดยหอคอยแรกเกิดจากแผ่นหิน N แผ่นที่มีรัศมี a[1], ..., a[N] มาเรียงกันจากล่างขึ้นบนตามลำดับ และ หอคอยที่สองเกิดจากแผ่นหิน M แผ่นที่มีรัศมี b[1], ..., b[M] มาเรียงกันจากล่างขึ้นบน ตามลำดับ เนื่องในโอกาสเฉลิมฉลองครบรอบวันเกิดของเมือง เจ้าเมืองจึงอยากจะทำการปรับปรุงหอคอยสองหอ ให้เป็นหอคอยคู่ นี้โดยการเอาแผ่นหินบางก้อนออก เพื่อให้หอคอยทั้งสองหอมี รูปร่างเหมือนกัน (คือมีความสูงเท่ากัน และรัศมีของแต่ละชั้น เท่ากัน) และสูงที่สุดเท่าที่จะเป็นไปได้ จงเขียน recurrence ของฟังก์ชั่นที่ใช้ในการคำนวณว่า หอคอยคู่นี้จะสูงที่สุดเท่าไรโดย ให้ M(i,j) แทนความสูงที่สุดที่เป็นไปได้ของหอคอยคู่ ที่เกิดจากหอคอยของแผ่นหิน a[1], ..., a[i] เรียงกันจากล่างขึ้นบน และ หอคอยของแผ่นหิน b[1],...,b[j] เรียงกันจากล่างขึ้นบนตามลำดับ และเขียน Pseudo Code ของการแก้ปัญหานี้แบบ bottom-up โดยสมมติว่าตาราง คือ M[i][j] พร้อมทั้งวิเคราะห์ประสิทธิภาพ (hint: อย่าลืม initial condition ของ recurrence)

- 7. (15 คะแนน) ปัญหาการแบ่งเท่า, ให้ตัวเลขจำนวนเต็มบวก a[1], a[N] โดยที่แต่ละตัวอยู่ระหว่าง 1 ถึง Q (1 <= a[i] <= Q) โดยเราอยากหาวิธีที่จะแบ่งตัวเลขเหล่านี้ออกเป็นสองกลุ่ม โดยให้ S1 เป็นผลรวมของ a ที่อยู่ในกลุ่ม 1 และ S2 เป็นผลรวมของ a ที่อยู่ในกลุ่ม 2 เราอยากจะทราบว่าจะแบ่งกลุ่มอย่างไรให้ | S1 S2 | น้อยที่สุดเท่าที่จะเป็นไปได้ ปัญหาข้อนี้ สามารถแก้ได้ ด้วยวิธี Dynamic Programming จงออกแบบฟังก์ชัน recurrence และเขียน recurrence ดังกล่าว พร้อมทั้งแสดงให้เห็นว่ามี ปัญหาย่อยที่ซ้ำกัน และให้วิเคราะห์ประสิทธิภาพของอัลกอริทึมที่สร้างขึ้นตาม recurrence ดังกล่าว (hint: อัลกอริทึมของคุณ ควรจะใช้เวลา และ หน่วยความจำเป็น polynomial ของ N และ Q และอย่าลืม initial condition ของ recurrence)
- 8. (15 คะแนน) มีเขาวงกตแห่งหนึ่งประกอบด้วยห้องเล็ก ๆ ต่อกันเป็นตารางจำนวน N * M ห้องกำหนดให้ห้องแต่ละห้องมีพิกัด กำกับเป็น (x,y) โดยที่ห้องมุมบนซ้ายคือห้องหมายเลข (0,0) และห้องมุมล่างขวาคือห้องหมายเลข (N-1,M-1) เราสามารถเดินทาง จากห้องหนึ่งไปยังอีกห้องหนึ่งในเขาวงกตนี้ได้ โดยมีกฎคือถ้าเราอยู่ที่ห้อง (X,Y) ใด ๆ เราจะสามารถเดินทางไปยังห้องที่อยู่ติดกัน คือห้อง (X+1,Y) หรือห้อง (X,Y+1) ได้เท่านั้น อย่างไรก็ตาม มีห้องบางห้องที่ "ห้ามเข้า" คือไม่สามารถเดินเข้าไปในห้องนั้นได้เลย กำหนดให้ A[x][y] เป็นตัวแปรที่บอกว่าห้องใดห้ามเข้าบ้าง โดยที่ A[x][y] จะมีค่าเป็น 1 ก็ต่อเมื่อห้องดังกล่าวห้ามเข้า สมมติให้ เราอยู่ที่ห้อง (0,0) และต้องการเดินทางไปยังห้อง (N-1,M-1) อยากทราบว่ามีวิธีการเดินที่แตกต่างกัน ทั้งหมดกี่แบบ (วิธีการเดิน จะแตกต่างกันกั้นกั้งหมายเลข (x,y) และต้องการเดินทางไปยังห้อง (N-1,M-1) แล้วจะมีวิธีการเดินที่แตกต่างกันทั้งหมดกี่แบบ และให้ ตอบว่าการนำเอา recurrence นี้ไปใช้เขียนโปรแกรมจะทำให้ได้โปรแกรมที่มีประสิทธิภาพในการทำงานเป็นอย่างไร (hint: ข้อนี้ ไม่จำเป็นต้องออกแบบอัลกอริทึม และอย่าลืม initial condition ของ recurrence)
- 9. (15 คะแนน) เราต้องการขับรถจากเชียงใหม่ไปสุไหง-โกลก แต่เนื่องจากระยะทางไกลมากเราจึงต้องแวะพักข้างทางอยู่เรื่อย ๆ สมมติให้มีจุดแวะพักทั้งหมด N จุด (แต่ละจุดมีหมายเลขกำกับ โดยจุดแรกคือหมายเลข 1 และให้ถือว่าเชียงใหม่และสุไหง-โกลก เป็นจุดพักหมายเลข 0 และหมายเลข N+1) การแวะพักที่จุดหมายเลข i นั้นจะทำให้เรามีแรงที่จะขับรถไปได้อีก r[i] จุดพัก (แรง ขับรถนี้ไม่มีการบวกทดกัน) ตัวอย่างเช่น แวะพักที่จุด 5 โดยที่ r[5] มีค่า 4 เราจะสามารถขับรถไปถึงจุด 6, 7, 8, 9 ได้ แต่ไม่ สามารถไปถึงจุด 10 ได้ ต้องแวะพักที่ใดที่หนึ่งระหว่างจุด 6 ถึง 9 ก่อน และถ้าหลังจากนั้นเราเลือกพักที่จุด 6 ต่อจากจุด 5 โดยที่ r[6] มีค่าเป็น 2 นั้น จะทำให้เราสามารถขับรถไปถึงจุดที่ 7 และ 8 ได้เท่านั้น ไม่สามารถไปถึงจุด 9 ได้ เพราะถือว่าการพักที่จุด 6 ทำให้เรามีแรงวิ่งไปได้แค่ 2 จุดพักเท่านั้น

การแวะพักจุด i นั้นจะต้องเสียเงินเป็นจำนวน p[i] บาท กำหนดให้ p[0] = 0 และให้คิดว่าเราแวะพักที่เชียงใหม่เรียบร้อยก่อน ออกเดินทาง กำหนดให้ 0 < r[i] <= N+1 จงออกแบบอัลกอริทึมเพื่อหาว่าเราจะต้องเสียเงินน้อยสุดเท่าไรเพื่อไปให้ถึงสุไหง-โกลก พร้อมทั้งวิเคราะห์ประสิทธิภาพในการทำงาน (hint: ไม่มี hint ครับ ขอให้โชคดี)