

FACULTY OF ENGINEERING
CHULALONGKORN UNIVERSITY
2110327 Algorithm Design

YEAR III, First Semester, Mid-term Examination, July 24, 2013, Time 8:30 – 11:30

ชื่อ-นามสกุล _____ เลขประจำตัว

									2	1
--	--	--	--	--	--	--	--	--	---	---

 CR58 _____

หมายเหตุ

1. ข้อสอบมีทั้งหมด 9 ข้อในกระดาษคำถามคำตอบจำนวน 8 แผ่น 8 หน้า คะแนนเต็ม 90 คะแนน
2. ไม่อนุญาตให้นำตำราและเครื่องคำนวณต่างๆ ใดๆ เข้าห้องสอบ
3. เขียนตอบในกระดาษข้อสอบชุดนี้ (ถ้าเนื้อที่ไม่พอ ให้เขียนต่อเฉพาะที่หน้าหลังของกระดาษคำถามแผ่นนั้น)
4. ควรเขียนตอบด้วยลายมือที่อ่านง่ายและชัดเจน
5. ห้ามการหยิบยืมสิ่งใดๆ ทั้งสิ้น จากผู้สอบอื่นๆ เว้นแต่ผู้คุมสอบจะหยิบยืมให้
6. ห้ามนำส่วนใดส่วนหนึ่งของข้อสอบออกจากห้องสอบ ข้อสอบเป็นทรัพย์สินของราชการซึ่งผู้ลักพาอาจมีโทษทางคดีอาญา
7. ผู้ที่ประสงค์จะออกจากห้องสอบก่อนหมดเวลาสอบ แต่ต้องไม่น้อยกว่า 45 นาที
8. เมื่อหมดเวลาสอบ ผู้เข้าสอบต้องหยุดการเขียนใดๆ ทั้งสิ้น
9. ผู้ที่ปฏิบัติเข้าข่ายทุจริตในการสอบ ตามประกาศคณะกรรมการการศึกษามิโทซ คือ ได้รับ ตัวย่อ F ในรายวิชาที่ทุจริตและพักการศึกษาอย่างน้อย 1 ภาคการศึกษา

รับทราบ

ลงชื่อนิติ (.....)

ต้องแสดงวิธีทำทุกข้อ การเขียนคำตอบเพียงอย่างเดียวจะไม่มีคะแนนให้ (ยกเว้นว่าจะเขียนในคำสั่ง)

1 (9 คะแนน) จงตอบคำถามต่อไปนี้สั้น ๆ ไม่ต้องอธิบาย

(ก) $2^{300} \bmod 11$ มีค่า = _____

(ข) $\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{n}$ = $\Theta(\text{_____})$

(ค) $5^1 + 5^2 + 5^3 + \dots + 5^n$ = $\Theta(\text{_____})$

(ง) $(\frac{1}{2})^1 + (\frac{1}{2})^2 + (\frac{1}{2})^3 + \dots + (\frac{1}{2})^n$ = $\Theta(\text{_____})$

(จ) การเรียงลำดับข้อมูล n ตัวด้วย mergesort ใช้เวลา = $\Theta(\text{_____})$

(ฉ) การ partition ใน quicksort ใช้เวลา = $\Theta(\text{_____})$

(ช) ให้ $T(n) = 9T(n/10) + n$, $T(0) = \Theta(1)$ จะได้ว่า $T(n) = \Theta(\text{_____})$

(ซ) 0/1 Knapsack คือปัญหาการหยิบของใส่ถุงโดยที่ถุงไม่ขาดและได้มูลค่ารวมสูงสุด ของมีหมายเลข $1, 2, \dots, n$ ของชิ้นที่ k มีน้ำหนัก w_k มีมูลค่า v_k และถุงจุของได้หนักไม่เกิน W จงเขียน recurrence ของมูลค่ารวมสูงสุดของการหยิบของใส่ถุง ในปัญหา 0/1 knapsack

2. (6 คะแนน) เป็นที่รู้กันทั่วไปว่า จำนวนฟีโบนัชชีหาได้จากนิยาม $F_n = F_{n-1} + F_{n-2}$, $F_0 = 0$, $F_1 = 1$ หากไปค้นใน Wikipedia จะพบว่า สามารถบรรยายค่าของจำนวนฟีโบนัชชีได้อีกมากมายหลายแบบ ดังตัวอย่างข้างล่างนี้ หากนำความรู้ของแต่ละแบบไปเขียนโปรแกรมหาค่าจำนวนฟีโบนัชชี อาจได้เวลาการทำงานต่างกัน จงวิเคราะห์เวลาการทำงานของโปรแกรมที่ใช้แต่ละรูปแบบข้างล่างนี้ในการคำนวณ (ไม่ต้องเขียนโปรแกรม) อธิบายที่มาของคำตอบด้วย (สั้นๆ)

	รูปแบบของจำนวนฟีโบนัชชีที่ใช้ในการคำนวณ	เวลาการทำงานของโปรแกรมที่คำนวณค่าของจำนวนฟีโบนัชชี (ให้ถือว่า การบวก ลบ คูณ หาร ใช้เวลาดังตัว)
1	$\begin{pmatrix} F_{k+2} \\ F_{k+1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} F_{k+1} \\ F_k \end{pmatrix}$ $\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n = \begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix}$	ตอบ
2	$F_{2n-1} = F_n^2 + F_{n-1}^2$ $F_{2n} = (2F_{n-1} + F_n)F_n$	ตอบ
3	$F_{3n} = 5F_n^3 + 3(-1)^n F_n$ $F_{3n+1} = F_{n+1}^3 + 3F_{n+1}F_n^2 - F_n^3$ $F_{3n+2} = F_{n+1}^3 + 3F_{n+1}^2F_n + F_n^3$	ตอบ

3. (5 คะแนน) สมชายไปพบแม่ที่อดภาษาจาวาดังแสดงด้านขวานี้ ที่หาค่าของ recurrence ะไรบางอย่างโดยใช้แนวคิดการทำงานแบบ bottom-up จงเขียน recurrence ของค่า $C(n)$ ที่เมที่อดนี้หา (เขียน initial condition ด้วย)

ตอบ

```
static int C(int n) {
    int[] C = new int[n + 1];
    C[0] = 1;
    for (int i = 1; i <= n; i++) {
        for (int k = 0; k < i; k++) {
            C[i] += C[k] * C[i - k - 1];
        }
    }
    return C[n];
}
```

4. (10 คะแนน)

- จงเขียน pseudo code ของอัลกอริทึม binary search

- จงเขียน pseudo code ของอัลกอริทึม quicksort (ไม่ต้องเขียน partition เรียกใช้ได้เลย)

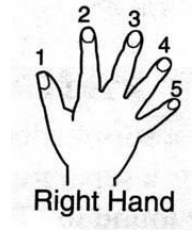
5. (10 คะแนน) ให้ **A** คืออาร์เรย์จัตุรัสสองมิติขนาด $n \times n$ ที่เก็บเลขจำนวนเต็ม (เป็นได้ทั้งบวกและลบ) จงเขียนการทำงานของ `valueOfMaxSquare(A[1..n][1..n])` แบบ pseudo code เพื่อบรรยายอัลกอริทึม ที่คืนผลรวมของอาร์เรย์ย่อยรูปจัตุรัสภายใน **A** ที่ผลรวมของทุกจำนวนในอาร์เรย์ย่อยนี้มีค่ามากที่สุด อัลกอริทึมที่นำเสนอนี้ต้องมีประสิทธิภาพเชิงเวลาเป็น $O(n^3)$ (ตัวอย่าง **A** ทางขวานี้ จะได้ค่า 40 เป็นผลลัพธ์)

0	0	0	1	1	2	1
0	1	0	-1	0	-1	0
1	-1	4	5	0	1	0
1	-1	8	0	1	1	-1
0	-5	-1	-1	9	3	-1
-9	-7	5	0	0	5	0
0	0	0	0	0	-5	0

ข้อแนะนำ: ควรสร้างอาร์เรย์ **S** ที่ `S[i][j]` เก็บผลรวมของข้อมูลในอาร์เรย์สี่เหลี่ยมย่อยตั้งแต่ `A[0][0]` ถึง `A[i][j]` (อาร์เรย์ตัวอย่างทางขวานี้คือ อาร์เรย์ **S** ของอาร์เรย์ **A** ด้านบน) เมื่อมี **S** การหาผลรวมอาร์เรย์สี่เหลี่ยมย่อยหนึ่ง ๆ ภายใน **A** จะสามารถทำได้ในเวลา $O(1)$ ลองคิดดู

0	0	0	1	2	4	5
0	1	1	1	2	3	4
1	1	5	10	11	13	14
2	1	13	18	20	23	23
2	-4	7	11	22	28	27
-7	-20	-4	0	11	22	21
-7	-20	-4	0	11	17	16

6.



(10 ค่ะแนน) ก่อนอื่นขอเรียกแต่ละแป้นกดของเปียโน (ทั้งขาวและดำ) ที่กดเป็นเสียงตัวโน้ตดังแสดงข้างบนนี้ว่า "คีย์" แต่ละคีย์จากซ้ายไปขวามีหมายเลขกำกับ 1, 2, ..., s (s คือจำนวนคีย์) เราต้องการบรรเลงเพลงด้วยการกดทีละคีย์ตามลำดับหมายเลขคีย์ที่เก็บในอาร์เรย์ $Note[1..n]$ ขอใช้เฉพาะนิ้วมือขวา กดครั้งละหนึ่งคีย์ (แต่ละนิ้วมีหมายเลขกำกับดังแสดงข้างบน) ปัญหาที่น่าสนใจในข้อนี้เรียกว่า *ปัญหาการวางนิ้วเล่นเปียโนที่ดีที่สุด (Optimal Piano Fingering)* เพื่อหาว่า จะใช้นิ้วใด กดคีย์ตามลำดับใน $Note$ เพื่อบรรเลงด้วย "ความยากรวม" น้อยสุด นั่นคือผลลัพธ์ที่ได้เป็นอาร์เรย์ $F[1..n]$ โดย $F[i]$ เก็บหมายเลขนิ้วที่ต้องกดคีย์ $Note[i]$ ที่มีความยากรวมน้อยสุด แต่โจทย์ข้อนี้เราไม่ต้องการให้หาอาร์เรย์ F หรือ ขอแค่เขียน recurrence บรรยายความยากรวมน้อยสุดก็พอ

กำหนดให้ฟังก์ชัน $d(f, k_1, g, k_2)$ คำนวณ "ความยากของการได้ใช้นิ้ว f กดคีย์ k_1 แล้วตามด้วยการใช้นิ้ว g กดคีย์ k_2 " ฟังก์ชัน d คำนวณจำนวนจริงแทนความยาก ตัวอย่างเช่น $d(1, 20, 2, 22)$ มีค่าน้อยกว่า $d(1, 10, 5, 88)$ เพราะนิ้วโป้งกดคีย์ 20 ตามด้วยนิ้วชี้กดคีย์ 22 ง่ายกว่า นิ้วโป้งกดคีย์ 20 ตามด้วยนิ้วนางกดคีย์ 88 ไม่ต้องคิดมาก ว่า ฟังก์ชันนี้ทำงานอย่างไร รู้แค่ว่า มันมีอยู่แล้ว ทำงานในเวลาคงตัว เราใช้ได้เลย

จงเขียน *recurrence* ของ $DP(j, f)$ แทน "ความยากรวมน้อยสุด" ของลำดับการใช้นิ้วเพื่อเริ่มกดคีย์ ตั้งแต่ $Note[j]$ จนถึง $Note[n]$ โดยเริ่มกดคีย์ $Note[j]$ ด้วยนิ้ว f (Hint: initial condition คือ $DP(n, f) = ?$)

(ดังนั้นความยากรวมน้อยสุดของการกดทั้งเพลงตั้งแต่ $Note[1]$ ถึง $Note[n]$ ก็คือ $\min_{1 \leq f \leq 5} \{ DP(1, f) \}$ n คือจำนวนคีย์ที่ต้องกดทั้งหมด)

7. (10 คะแนน) ให้อาร์เรย์ $Y[1..n]$ เก็บเลขปี พ.ศ. ที่มีการจัดการเลือกตั้งในประเทศหนึ่ง โดยที่ $Y[1] < Y[2] < \dots < Y[n]$ ปัญหาข้อนี้ต้องการหา $F(Y, x)$ ซึ่งคือจำนวนการเลือกตั้งที่มากที่สุดภายในช่วง x ปีใดๆที่ติดกัน
- ตัวอย่าง ให้ $Y = [2500, 2502, 2508, 2510, 2511, 2512, 2516, 2519]$
- $F(Y, 4)$ มีค่า 3 เนื่องจากในช่วงปี 2509 ถึง 2512 (4 ปีติดกัน) มีการเลือกตั้ง 3 ครั้ง ซึ่งมากกว่าหรือเท่ากับจำนวนการเลือกตั้งในช่วง 4 ปีที่ติดกันใดๆ
- ก) (8 คะแนน) จงเขียน pseudo code ของ $F(Y, x)$ ที่ทำงานแบบ Divide and Conquer ที่ใช้เวลา $O(n \log n)$

- ข) (2 คะแนน) จงวิเคราะห์เวลาการทำงานของรหัสเทียมในข้อ ก (นิสิตต้องเขียนอธิบายจึงจะได้คะแนน)

8. (15 คะแนน) ให้ \mathbf{a} เป็นอาร์เรย์ของจำนวนเต็มที่มีค่าไม่ซ้ำกันเลย เราเรียกคู่ข้อมูล $\mathbf{a}[i]$ กับ $\mathbf{a}[j]$ ว่า เกิดการสลับ (inversion) เมื่อ $i < j$ แต่ $\mathbf{a}[i] > \mathbf{a}[j]$ ข้อนี้ต้องการให้นิสิตหาวิธีนับจำนวนการสลับทั้งหมดในอาร์เรย์ เช่น $\mathbf{a} = [30, 10, 20, 50, 40]$ มีจำนวนการสลับทั้งหมดเป็น 3 จากคู่ข้อมูล (30,10), (30,20) และ (50,40)
- ก) (13 คะแนน) จงเขียน pseudo code เพื่อหาจำนวนการสลับทั้งหมดของอาร์เรย์ที่ได้รับ ด้วยวิธี Divide and Conquer ที่ใช้เวลาการทำงานเป็น $O(n \log n)$

ข) (2 คะแนน) จงวิเคราะห์การทำงานของ pseudo code ในข้อ ก (นิสิตต้องเขียนอธิบายจึงจะได้คะแนน)

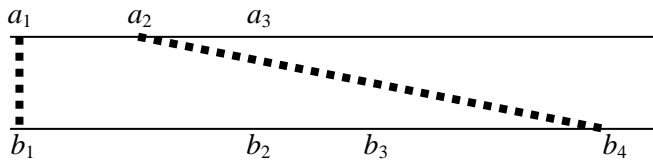
9. (15 คะแนน) สมมติว่านิสิตเป็นที่ปรึกษาด้านการวางผังเมืองให้กับรัฐบาลของประเทศไดนามิกแลนด์ ประเทศนี้มีแม่น้ำเพียงสายเดียว (แม่น้ำซี) ไหลจากชายแดนด้านตะวันตกไปยังชายแดนด้านตะวันออกของประเทศ และแบ่งประเทศเป็นฝั่งเหนือและใต้ ทางฝั่งเหนือมีเมืองที่ติดแม่น้ำอยู่ n เมืองอยู่ที่พิกัดทางแนวนอนคือ $a_1 < a_2 < \dots < a_n$ และทางฝั่งใต้ มีเมืองที่ติดแม่น้ำอยู่ m เมืองอยู่ที่พิกัดทางแนวนอน $b_1 < b_2 < \dots < b_m$ การสัญจรระหว่างเมืองคนละฝั่งแม่น้ำ ซึ่งใช้เรือ จึงไม่สะดวก แต่ทว่าในปีนี้ รัฐบาลของไดนามิกแลนด์ ได้อนุมัติงบประมาณเมกะโปรเจกต์อันมหาศาล เพื่อสร้างสะพานข้ามแม่น้ำให้ได้ประโยชน์สูงสุดเท่าที่ทำได้ เพื่อเพิ่มความสะดวกสบายให้แก่พี่น้องชาวไดนามิกแลนด์ แต่เนื่องด้วยข้อจำกัดทางวิศวกรรมและข้อจำกัดทางการเมือง สะพานที่จะสร้างได้จะต้องอยู่ภายใต้เงื่อนไขดังต่อไปนี้

- สะพานจะต้องเชื่อมเมืองหนึ่งทางฝั่งเหนือ ไปยังอีกเมืองหนึ่งยังฝั่งใต้เท่านั้น ไม่อนุญาตให้สร้างสะพานเชื่อมบริเวณที่ไม่ฝั่งใดฝั่งหนึ่งหรือทั้งสองฝั่งมิใช่เมือง (นั่นคือ ตำแหน่งปลายสะพาน ต้องอยู่ที่พิกัด a_1, \dots, a_n และ b_1, \dots, b_m)
- เมืองหนึ่งเมืองมีสะพานข้ามแม่น้ำได้ไม่เกินหนึ่งสะพานเท่านั้น
- ห้ามมีสะพานใด ๆ ตัดกันโดยเด็ดขาด (สะพานที่เชื่อม a_i กับ b_j จะตัดกับสะพานที่เชื่อม a_k กับ b_g ก็ต่อเมื่อ $((a_i < a_k) \text{ AND } (b_g < b_j)) \text{ OR } ((a_k < a_i) \text{ AND } (b_j < b_g))$)
- สะพานที่เชื่อมจาก a_i ไปยัง b_j จะก่อให้เกิดประโยชน์ที่วัดได้เป็นค่าเท่ากับ $B(i, j)$ (มีผู้เชี่ยวชาญคำนวณให้แล้ว)
- ประโยชน์ของสะพานทั้งหมด คือผลรวมของประโยชน์ของแต่ละสะพานที่ถูกสร้าง

ตัวอย่าง $a_1 = 5, a_2 = 10, a_3 = 15, b_1 = 5, b_2 = 15, b_3 = 20, b_4 = 30$ $B(i, j) =$

คำตอบคือ 110 (Bridge 1: a_1 to b_1 , Bridge 2: a_2 to b_4)

10	1	2	3
2	3	3	100
4	2	1	5



ก) (5 คะแนน) จงเขียน recurrence ของประโยชน์รวมสูงสุดที่เป็นไปได้ที่ไม่คิดเงื่อนไข

ข) (10 คะแนน) จงเขียน pseudo code เพื่อคำนวณประโยชน์รวมสูงสุดที่เป็นไปได้ และ แสดงเมืองที่สะพานแต่ละสะพานเชื่อม ออกมาทางหน้าจอ โดยอัลกอริทึมที่ยอมรับได้จะต้องใช้เวลาไม่เกิน $O(n^2m^2)$

(สำหรับ เขียนคำตอบของข้อ 9)