

FACULTY OF ENGINEERING
CHULALONGKORN UNIVERSITY
2110327 Algorithm Design

YEAR III, First Semester, Final Examination, September 24, 2009, Time 8:30 – 11:30

ชื่อ-นามสกุล _____ เลขประจำตัว

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

 CR58 _____

หมายเหตุ

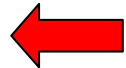
1. ข้อสอบมีทั้งหมด 15 ข้อในกระดาษคำถามคำตอบจำนวน 2 แผ่น 4 หน้า คะแนนเต็ม 80 คะแนน
2. ไม่อนุญาตให้นำคำราและเครื่องคำนวณต่างๆ ใดๆ เข้าห้องสอบ
3. ควรเขียนตอบด้วยลายมือที่อ่านง่ายและชัดเจน
4. ห้ามการหยิบยืมสิ่งใดๆ ทั้งสิ้น จากผู้สอบอื่นๆ เว้นแต่ผู้คุมสอบจะหยิบยืมให้
5. ห้ามนำส่วนใดส่วนหนึ่งของข้อสอบออกจากห้องสอบ ข้อสอบเป็นทรัพย์สินของราชการซึ่งผู้ลักพาอาจมีโทษทางคดีอาญา
6. ผู้ที่ประสงค์จะออกจากห้องสอบก่อนหมดเวลาสอบ แต่ต้องไม่น้อยกว่า 45 นาที
7. เมื่อหมดเวลาสอบ ผู้เข้าสอบต้องหยุดการเขียนใดๆ ทั้งสิ้น
8. ผู้ที่ปฏิบัติเข้าข่ายทุจริตในการสอบ ตามประกาศคณะกรรมการฯ มีโทษ คือ ได้รับ สัญลักษณ์ F ในรายวิชาที่ทุจริต และพักการศึกษาอย่างน้อย 1 ภาคการศึกษา

รับทราบ

ลงชื่อนิสิต (.....)

หมายเหตุ (เพิ่มเติม)

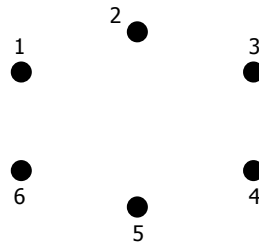
1. ข้อใดที่ให้ออกแบบอัลกอริทึมนั้น นิสิตสามารถตอบโดยเขียนบรรยายแนวคิดที่ **implement** ได้ในทางปฏิบัติ หรือจะเขียนเป็นรหัสเทียมประกอบแนวคิดที่นำเสนอด้วยก็ได้ และต้องวิเคราะห์ประสิทธิภาพเชิงเวลาของอัลกอริทึมที่นำเสนอด้วย
2. ต้องแสดงวิธีทำทุกข้อ การเขียนคำตอบเพียงอย่างเดียวจะไม่มีคะแนนให้ (ยกเว้นว่าจะเขียนในคำสั่ง)
3. ให้นิสิตเขียนรหัสประจำตัวและเลขที่ใน CR58 ในทุกหน้าของกระดาษคำถามด้วย
4. ข้อสอบนี้มีสองส่วน ส่วนแรกให้เขียนคำตอบในกระดาษข้อสอบ ส่วนที่ 2 ให้เขียนตอบในสมุดคำตอบ โดยเขียน **ตอบข้อที่ k ไว้ที่หน้า k ในสมุดคำตอบ** ($k = 1, 2, \dots, 5$) เขียนไม่พอ ไปต่อที่หน้า 6, ...



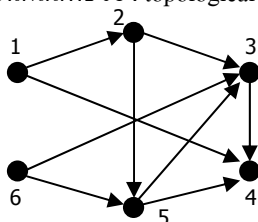
ส่วนที่ 1 ตอบคำถามข้างล่างนี้ในช่องว่างของแต่ละคำถามต่อไปนี้

1. (3 คะแนน) จงวาดเส้นเชื่อม ทิศทาง และน้ำหนักในกราฟทางขวา ให้สอดคล้องกับ adjacency matrix ทางซ้ายนี้

	1	2	3	4	5	6
1	∞	5	∞	∞	∞	-1
2	∞	∞	-3	∞	7	∞
3	∞	∞	∞	4	∞	∞
4	9	∞	∞	∞	-2	∞
5	∞	∞	∞	∞	∞	6
6	∞	∞	8	∞	∞	∞



2. (3 คะแนน) จงเขียนผลลัพธ์ของ topological sort (หรือที่เรียกว่า linearization of graph) ทุกรูปแบบของกราฟข้างล่างนี้



ตอบ

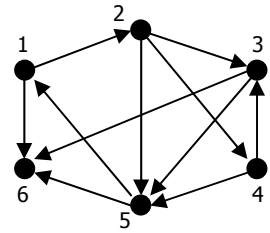
.....

.....

.....

.....

3. (3 คะแนน) ถ้าทำ breadth-first search ในกราฟทางขวานี้ โดยเริ่มที่ปมหมายเลข 1 จะต้องใช้ queue (แถวคอย) ที่สามารถเก็บปมได้อย่างน้อยเท่าใดระหว่างการค้น จึงจะทำ breadth-first search ได้สำเร็จ ตอบ _____ ปม



4. (3 คะแนน) อัลกอริทึมของ Dijkstra มีไว้หาเส้นทางสั้นสุดจากปมต้นทาง s ไปยังปมอื่นในกราฟ จงบรรยายวิธีการหาเส้นทางสั้นสุดจากทุก ๆ ปมในกราฟมายังปมปลายทาง t ของกราฟ ในเวลาที่พอ ๆ กับของ Dijkstra (บรรยายสั้น ๆ 2-3 บรรทัด)

.....

.....

.....

5. (3 คะแนน) กำหนดให้ G คือกราฟต่อถึงกัน (connected) ที่เส้นเชื่อมไม่มีทิศทาง เส้นเชื่อมมีความยาว (ติดลบได้) และเป็นกราฟที่ไม่มีวงจร อยากทราบว่าวิธีใดต่อไปนี้ใช้หาเส้นทางสั้นสุดระหว่างสองปมที่กำหนดให้ในกราฟนี้ได้ (วงชื่อวิธีทั้งหมดที่ใช้ได้)

depth-first search breadth-first search Dijkstra's algorithm ใช้ไม่ได้เลยสักวิธี

6. (3 คะแนน) กำหนดให้ G เป็น undirected weighted graph ถ้าเราสร้าง graph H โดยให้ H คือ G ที่ทุก ๆ เส้นเชื่อมในกราฟนั้นถูกเพิ่ม weight ด้วยค่าคงที่ k ผลลัพธ์ของการใช้ Kruskal's Algorithm และ Dijkstra Algorithm บนกราฟ G และ H ต่างกันหรือไม่? (ให้ตอบเพียงว่า ต่าง หรือ ไม่ต่าง)

Kruskal's Algorithm _____

Dijkstra's Algorithm _____

7. (3 คะแนน) นิสิตคนหนึ่งเขียนปฏิบัติการที่ทำ state space search กับปัญหา Knapsack ดังแสดงข้างล่างนี้ จงวาด state space tree เมื่อใช้ฟังก์ชัน knapsack นี้กับของ 3 ชิ้นเก็บในอาร์เรย์ items ที่มี (มูลค่า, น้ำหนัก) เป็น (20, 8), (40, 5), (10, 4), และ อนุญาตได้ 10 ($W = 10$)

```
int totalValue(int x[], int k) {
    int v = 0, i = 0;
    for(i=0; i<k; i++) v+=items[i].value*x[i];
    return v;
}
int totalWeight(int x[], int k) {
    int w = 0, i = 0;
    for(i=0; i<k; i++) w+=items[i].weight*x[i];
    return w;
}
//-----
void knapsack(int x[], int k) {
    if (totalWeight(x, k) > W) return;
    if (k == n) {
        if (totalValue(x, k) > maxValue) {
            maxValue = totalValue(x);
            int i = 0;
            for(i=0; i<n; i++) maxSol[i] = x[i];
        }
    } else {
        x[k] = 1; knapsack(x, k+1);
        x[k] = 0; knapsack(x, k+1);
    }
}
```

วาด state-space tree ดังนี้

8. (3 คะแนน) ปัญหา SPJ เป็นดังนี้ “ให้ G เป็น directed acyclic graph อยากทราบว่า G มี directed path ที่ผ่านทุกปม ปมละ 1 ครั้งหรือไม่” จงบรรยายสั้น ๆ เพื่อพิสูจน์ว่าปัญหา SPJ อยู่ในกลุ่มปัญหา NP

.....

.....

.....

9. (3 คะแนน) จงบรรยายสั้น ๆ เพื่อพิสูจน์ว่าปัญหา SPJ (ในข้อที่แล้ว) อยู่ในกลุ่มปัญหา P หรือไม่ก็พิสูจน์ว่า SPJ อยู่ในกลุ่มปัญหา NP -complete (เลือกพิสูจน์หนึ่งอย่าง)

.....

.....

.....

10. (3 คะแนน) ทางขวาคือตารางที่ได้จากการหาคำตอบของ LCS(x, y) ด้วยวิธี dynamic programming โดยมี $x = x_1x_2x_3x_4x_5$ และ $y = y_1y_2y_3y_4$ อยากทราบว่า ผลลัพธ์การจับคู่ตัวใดของ x ตรงกับตัวใดของ y ตอบ

	x_1	x_2	x_3	x_4	x_5
0	0	0	0	0	0
y_1	0	1	1	1	1
y_2	0	1	2	2	2
y_3	0	1	2	2	2
y_4	0	1	2	2	3

ส่วนที่ 2 ตอบคำถามข้างล่างนี้ในช่องว่างของแต่ละคำถามต่อไปนี้ โดยให้ตอบข้อที่ k ไว้ที่หน้าที่ k ในสมุดคำตอบ

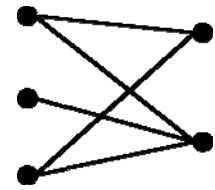
1. (10 คะแนน) ปัญหานี้เป็นปัญหาการหาทางเดินสั้นสุด (Shortest Path) สมมติให้มี undirected weighted connected graph $G = (V, E)$ ซึ่งมีปมอยู่ n ปม และมี $A[][]$ เป็นอาร์เรย์ 2 มิติ ที่เก็บ adjacency matrix ของ graph ดังกล่าวโดยที่ $A[i][j]$ (และ $A[j][i]$) จะเก็บ weight จากปม i ไปยังปม j และจะเก็บค่า 0 ถ้าไม่มีเส้นเชื่อมจากปม i ไปยังปม j และสมมติให้ $D[][]$ เป็นอาร์เรย์ 2 มิติที่เก็บผลลัพธ์ของการหาระยะทางสั้นสุดของทุก ๆ คู่ปมโดยที่ตัวแปร D นั้นถูกคำนวณไว้เรียบร้อยแล้ว จงออกแบบอัลกอริทึมที่ใช้เวลา $O(|V|^2)$ โดยเขียน **pseudo-code** ที่ทำการหา shortest path จากปม p ไปยังปม q ตามโครงฟังก์ชันต่อไปนี้ (นิสิตอาจจะเขียนด้วยภาษาอื่นก็ได้) ให้เก็บผลลัพธ์ลงในอาร์เรย์ path และให้ฟังก์ชัน return จำนวนปมใน path (ต้องวิเคราะห์ประสิทธิภาพการทำงานด้วย)

```
int FindPath(int A[ ][ ], int D[ ][ ], int n, int p, int q, int path[ ]) {
    // insert your pseudo-code here
}
```

2. (10 คะแนน) ปัญหาเกมหีบหินเป็นดังนี้ สมมติให้มีหินอยู่ $2N$ ก้อนเรียงเป็นแถว หินแต่ละก้อนมีน้ำหนักต่าง ๆ กันไป (น้ำหนักอาจจะเท่ากันได้) ผู้เล่นสองคนจะผลัดกันหีบหินทีละก้อน โดยผู้เล่นแต่ละคนจะหีบหินได้เฉพาะก้อนที่อยู่หัวแถว หรืออยู่ปลายแถวเท่านั้น เมื่อหีบแล้ว ให้เอาหินก้อนนั้นมาเก็บไว้กับตัว เกมจะจบลงเมื่อผู้เล่นทั้งคู่ได้หีบหินไปครบ N ก้อน ผู้ชนะคือผู้ที่มีน้ำหนักรวมของหินมากที่สุด

- ก. สมมติให้มีหินสี่ก้อน ดังนี้ 20, 19, 1, 3 จงเขียน state space tree ของการหีบหินที่เป็นไปได้ทั้งหมด
- ข. จงออกแบบอัลกอริทึมแบบ branch-and-bound เพื่อหาว่าน้ำหนักรวมมากที่สุดที่ผู้เล่นคนแรกจะหีบได้เป็นเท่าไร (ไม่ต้องวิเคราะห์ประสิทธิภาพการทำงาน) พร้อมทั้งอธิบาย bound ที่ใช้ในอัลกอริทึมด้วย

3. (10 คะแนน) กราฟสองฝ่าย (Bipartite Graph) คือ undirected graph $G = (V, E)$ ที่ปมในกราฟนี้สามารถถูกแบ่งออกเป็น 2 ฝ่ายโดยที่ไม่มีเส้นเชื่อมระหว่างปมที่อยู่ในฝ่ายเดียวกัน (ตัวอย่างดังรูปด้านขวามือ) กราฟถูกแบ่งออกเป็นปมฝ่ายซ้าย 3 ปม และ ปมฝ่ายขวา 2 ปม) จงออกแบบอัลกอริทึมที่ใช้เวลา $O(|V| + |E|)$ สำหรับตรวจสอบว่ากราฟ G เป็นกราฟสองฝ่ายหรือไม่ (ต้องวิเคราะห์ประสิทธิภาพการทำงานด้วย)



4. (10 คะแนน) จงเขียนฟังก์ชันภาษา C ของอัลกอริทึมของ Prim เพื่อทำการหา Minimal Spanning Tree (MST) ของกราฟ $G = (V, E)$ โดยกำหนดให้ G เป็น undirected weighted complete graph ซึ่งมีปมอยู่ n ปม และมี $A[][]$ เป็นอาร์เรย์ 2 มิติ ที่เก็บ adjacency matrix ของ graph ดังกล่าว โดยที่ $A[i][j]$ (และ $A[j][i]$) จะเก็บ weight จากปม i ไปยังปม j ฟังก์ชันที่เขียนขึ้นนั้นจะต้องเก็บผลลัพธ์ไว้ในอาร์เรย์ parent ซึ่งจะบอกถึงปมพ่อ (parent node) ของแต่ละปมใน MST และกำหนดให้ปมรากมีปมพ่อเป็น -1 ให้ใช้โครงฟังก์ชันดังนี้

```
void MST(int A[ ][ ], int n, int parent[ ]) {
    // insert your code here
}
```

5. (10 คะแนน) ปัญหาซudoku หรือเกมเติมตัวเลขมิกติกาคือ เติมตัวเลข 1 ถึง 9 ในแต่ละช่องของตารางหลักขนาด 9×9 ช่อง ซึ่งประกอบด้วยตารางย่อย 3×3 ช่อง จำนวน 9 ตาราง ตัวเลขบางตัวแสดงไว้แล้วในบางช่อง ตัวเลขแต่ละตัวจะมีได้แค่หนึ่งตำแหน่งเท่านั้น ในแต่ละแถว ในแต่ละสดมภ์ และในแต่ละตารางย่อย และ หนึ่งช่องมีตัวเลขได้เพียงตัวเดียว (ตัวอย่างด้านล่างคือผลเฉลยหนึ่งของปัญหาซudoku) จงเขียนอัลกอริทึมลดรูป (reduce) ปัญหาซudoku 9×9 ให้เป็น ปัญหา SAT (เปลี่ยนเป็นนิพจน์ boolean) ในเวลาพหุนาม (ต้องวิเคราะห์ประสิทธิภาพการทำงานด้วย)

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9