ชื่อ _____ รหัส _____ section ___

สอบปลายภาค: ภาคการศึกษาที่ 2 **ปีการศึกษา**: 2557

วันสอบ: 18 พฤษภาคม 2558 **เวลาสอบ**: 13.30 – 16.30 น.

ห้องสอบ: A200, A401

ผู้สอน: อ.เสกสรรค์ สุวรรณมณี อ.อภิชาติ หีดนาคราม ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

รหัสและชื่อวิชา: 242-310 Introduction to Algorithm and Complexity

แนะนำขั้นตอนวิธีและความซับซ้อน

ทุจริตในการสอบมีโทษขั้นต่ำคือ ปรับตกในรายวิชาที่ทุจริตและพักการเรียน ๒ ภาคการศึกษา

คำสั่ง: **อ่านรายละเอียดของข้อสอบ และคำแนะนำให้เข้าใจก่อนเริ่มทำข้อสอบ**

อนุญาต: เครื่องเขียนต่างๆ เช่น ปากกา หรือดินสอ เข้าห้องสอบ และ กระดาษขนาด A4 หนึ่งแผ่น จดบันทึกด้วยลายมือเท่านั้น (ห้าม print หรือ ถ่ายเอกสาร)

ไม่อนุญาต: หนังสือ หรือเครื่องคิดเลขเข้าห้องสอบ และ**เอกสารใดๆ เข้าและออกห้องสอบ**

เวลา: 3 ชั่วโมง (180 นาที)

คำแนะนำ

- ☐ ข้อสอบมี 13 **หน้า** (*รวมหน้าปก*) แบ่งออกเป็น 5 **ตอน** คะแนนรวม 100 คะแนน (คิดเป็นคะแนน 35%)
- ☐ **เขียนคำตอบในข้อสอบ คำตอบส่วนใดอ่านไม่ออก จะถือว่าคำตอบนั้นผิด**
- ☐ อ่านคำสั่งในแต่ละข้อให้เข้าใจก่อนลงมือทำ
- ☐ เวลาที่ใช้เวลาทำตอนให้เหมาะสม ตามคำแนะนำ
- ☐ หากข้อใดเขียนคำตอบไม่พอ ให้เขียนเพิ่มที่ด้านหลังของหน้านั้นเท่านั้น
- ☐ **ให้ส่งกระดาษจดโน้ตพร้อมข้อสอบ** เขียนชื่อ-รหัสนักศึกษา ให้ชัดเจน

ตอน	**1** (25)	**2** (20)	**3** (25)	**4** (20)	**5** (10)	**รวม** (100) 35%
คะแนน						

นักศึกษารับทราบ ลงชื่อ .....

ตอนที่ 1 (25 คะแนน, 50 นาที)

กำหนดการพลวัต (Dynamic Programming)

1. การทำงานแบบ Top-Down และ Bottom-Up มีลักษณะอย่างไร จงอธิบายความแตกต่าง (5 คะแนน)

2. จงอธิบายความหมายของคำสำคัญต่อไปนี้ (5 คะแนน)

1) Feasible Solution

2) Objective Function

3) Optimal Solution

4) Constraints

5) Optimal Substructure

3. ปัญหา 0/1 Knapsack กำหนดให้มีของจำนวน n ชิ้น แต่ละชิ้นมีมูลค่า $v_1, v_2, \ldots, v_n$ และน้ำหนัก $w_1, w_2, \ldots, w_n$ ตามลำดับ ถุงเป้หนึ่งใบจุของได้หนักไม่เกิน W จงเลือกของใส่ถุงให้ได้มูลค่ารวมมากสุด โดยที่ถุงไม่ขาด

Objective function (ฟังก์ชันจุดประสงค์):

$$\text{Maximize} \quad v_{total} = \sum_{i=1}^{n} (x_i * v_i)$$

Constraint (ข้อจำกัดหรือเงื่อนไข):

$$w_{total} = \sum_{i=1}^{n} (x_i * w_i) \leq W$$

1) จงเขียน recurrence relation ของปัญหานี้ (3 คะแนน)

2) จงแสดงการเติมตารางด้านล่าง โดยใช้ของจำนวน 4 ชิ้น แต่ละชิ้นมีมูลค่า 40, 15, 30, 20 บาท และน้ำหนัก 2, 2, 3, 2 กิโลกรัม ตามลำดับ ถุงเป้รับน้ำหนักได้ 6 กิโลกรัม (5 คะแนน)

	j=0	1	2	3	4	5	6
i=0	0	0	0	0	0	0	0
1	0						
2	0						
3	0						
4	0						

3) จงเขียนอัลกอริทึมสำหรับหามูลค่ารวมมากที่สุดของ ปัญหา 0/1 Knapsack โดยใช้หลักการ Dynamic Programming (bottom-up approach) พร้อมอธิบาย Code (7 คะแนน)

/********** **จบตอนที่ 1** **********/

ตอนที่ 2 (20 คะแนน, 30 นาที)

อัลกอริทึมเชิงละโมบ (Greedy Algorithm)

1. จงตอบว่าประโยคต่อไปนี้ จริง หรือ เท็จ พร้อมให้เหตุผลสนับสนุน หรือ โต้แย้ง (5 คะแนน)

1) อัลกอริทึมเชิงละโมบรับประกันคำตอบที่ดีที่สุดเสมอ

คำตอบ:

เหตุผล:

2) การหาระยะทางที่สั้นที่สุดสามารถใช้อัลกอริทึมเชิงละโมบได้

คำตอบ:

เหตุผล:

3) ในปัญหา fractional knapsack เราควรเลือกของที่แพงที่สุดก่อน

คำตอบ:

เหตุผล:

4) ปัญหา 0/1 knapsack ไม่สามารถใช้อัลกอริทึมเชิงละโมบ

คำตอบ:

เหตุผล:

5) ปัญหาการทอนเงินโดยใช้อัลกอริทึมเชิงละโมบอาจนำไปสู่คำตอบที่ผิดได้

คำตอบ:

เหตุผล:

2. ปัญหาการเลือกกิจกรรม Activity Selection (15 คะแนน)

มีกิจกรรม n อย่าง (A_1, A_2, ..., A_n) ที่ต้องการจัดให้ห้องๆหนึ่ง มีการจองเวลาใช้งานเริ่มต้นและสิ้นสุดแต่ละกิจกรรมเป็น s_1, s_2 ,..., s_n และ f_1, f_2, ..., f_n จะเลือกกิจกรรมอย่างไรให้ได้จำนวนกิจกรรมมากที่สุด โดยที่ใช้ห้องไม่เหลื่อมล้ำกัน (เช่น ถ้า $f_j \leq s_i$ สามารถจัดกิจกรรม i และ j ได้)

1) จงอธิบายหลักการแก้ปัญหา Activity selection ด้วย Greedy algorithm พร้อมอธิบาย Code

2) จากข้อมูลตัวอย่างปัญหาการเลือกกิจกรรม ดังข้างล่างนี้ (ข้อมูลเวลาเริ่มต้นและสิ้นสุดของ 10 กิจกรรม A_1, A_2, ..., A_{10}) จงหาว่าจำนวนกิจกรรมสูงสุดที่เลือกได้เป็นเท่าไร และต้องเลือกกิจกรรมใดบ้าง จงแสดงวิธีทำอย่างละเอียด

Activity	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}
Start Time	10	7	3	5	2	8	6	2	4	3
Finish Time	14	8	7	6	3	11	7	6	5	8

/********** **จบตอนที่** 2 **********/

ตอนที่ 3 (25 คะแนน, 50 นาที)

อัลกอริทึมกราฟ (Graph Algorithms)

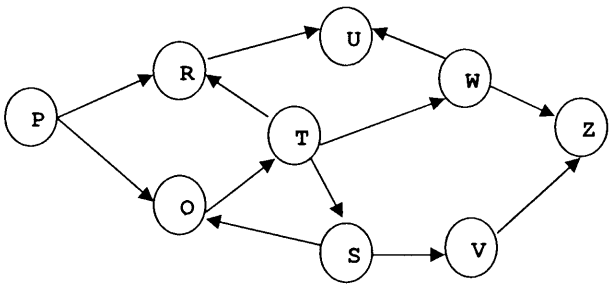
1. Complete Graph ที่มี n ปมจะมีจำนวนเส้นเชื่อมเท่าไร (1 คะแนน)

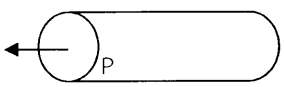
2. Undirected Graph, Directed Graph ต่างกันอย่างไร (1 คะแนน)

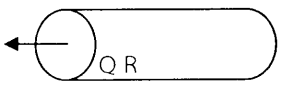
3. Path และ Cycle ต่างกันอย่างไร (1 คะแนน)

4. Graph และ Tree ต่างกันอย่างไร (1 คะแนน)

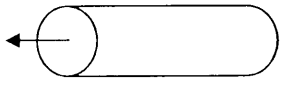
5. จงใช้อัลกอริทึมสำหรับการค้นตามแนวกว้าง (Breadth First Search) แสดงการไหลเข้าออก Queue ของแต่ละปม ทีละขั้นตอน โดยใช้ Graph และ Queue ที่ให้มาต่อไปนี้ (10 คะแนน)



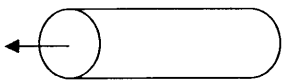
1)  (ตัวอย่าง)

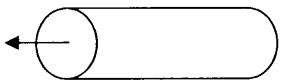
2)  (ตัวอย่าง)

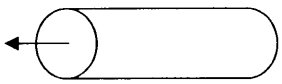
3)

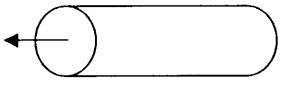
4) 

5)

6) 

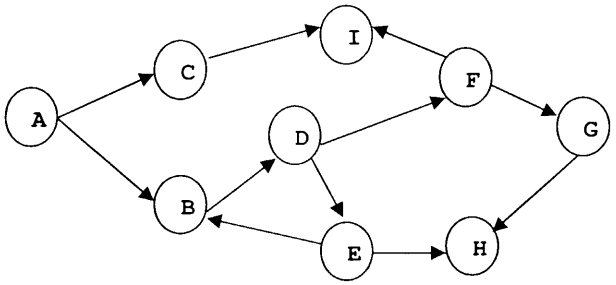
7) 

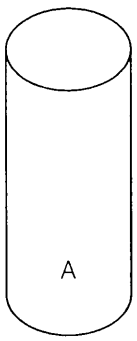
8) 

9) 

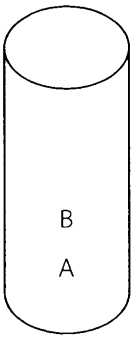
6. จากข้อ 5 หากกำหนดให้ทุกเส้นมียาวเท่ากับ 1 วิถีสั้นสุดจาก P ไปหา Z มีความยาวเป็นเท่าใด (1 คะแนน)

7. จงใช้อัลกอริทึมสำหรับการค้นตามแนวลึก (Depth First Search) แสดงการไหลเข้าออก Stack ของแต่ละปม ทีละขั้นตอน โดยใช้ Graph และ Stack ที่ให้มาต่อไปนี้ (10 คะแนน)

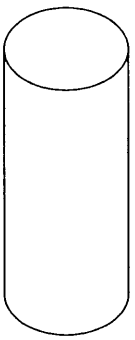


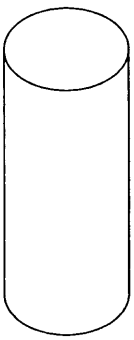
1) 

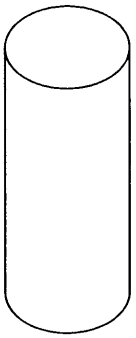
(ตัวอย่าง)

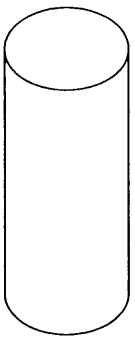
2) 

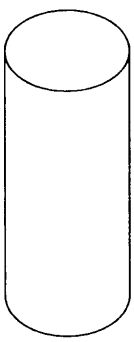
(ตัวอย่าง)

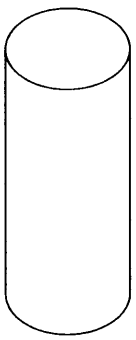
3) 

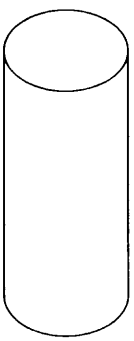
4) 

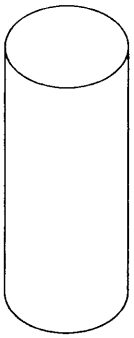
5) 

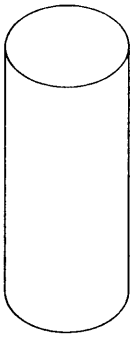
6) 

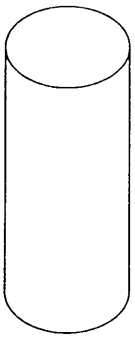
7) 

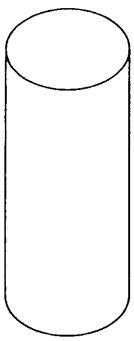
8) 

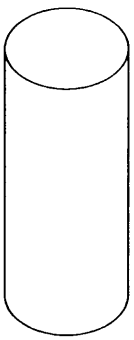
9) 

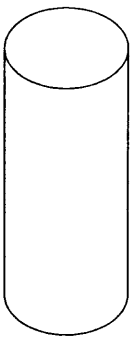
10) 

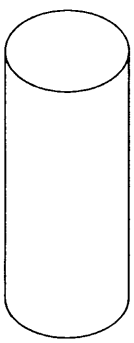
11) 

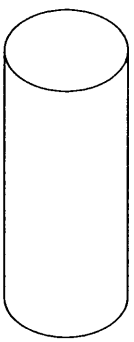
12) 

13) 

14) 

15) 

16) 

17) 

/********** **จบตอนที่ 3** **********/

ตอนที่ 4 (20 คะแนน, 30 นาที)

การค้นในปริภูมิสถานะ (State Space Search)

1. ใช้ตัวเลือกต่อไปนี้ในการตอบคำถามด้านล่าง (6 คะแนน)

A. Depth-First Search B. Breadth-First Search

C. Backtracking D. Least-cost Search

1) ในปริภูมิสถานะ (state space) ที่เป็น binary tree จากวิธีการค้นปริภูมิสถานะ(state space search) เริ่มจากซ้ายมือ วิธีการใดมีประสิทธิภาพต่ำที่สุดเมื่อคำตอบของปัญหาอยู่ที่กิ่งทางขวามือสุด เพราะเหตุใด

คำตอบ:

เหตุผล:

2) ในปริภูมิสถานะ (state space) ที่เป็น binary tree จากวิธีการค้นปริภูมิสถานะ(state space search) วิธีการใดมีประสิทธิภาพต่ำที่สุดเมื่อคำตอบของปัญหาอยู่ที่โหนดใบลึกสุด เพราะเหตุใด

คำตอบ:

เหตุผล:

2. จงวาดปริภูมิสถานะ และสาธิตการค้นหาด้วย Least-cost Search สำหรับ Sum of Subset Problem เมื่อ A = {12, 8, 7, 2} และ ผลรวม P = 22 (6 คะแนน)

3. จงวาดปริภูมิสถานะ และสาธิตการค้นหาด้วย Backtracking สำหรับ Sum of Subset Problem เมื่อ A = {12, 8, 7, 2} และ ผลรวม P = 22 (6 คะแนน)

4. จากข้อ 2 และข้อ 3 วิธีการใดค้นพบคำตอบได้เร็วกว่า (2 คะแนน)

/********** **จบตอนที่ 4** **********/

ตอนที่ 5 (10 คะแนน, 20 นาที)

ความซับซ้อนของปัญหา (Complexity Class)

1. จงแปลง Optimization Problems ต่อไปนี้ให้อยู่ในรูปของ Decision Problems (3 คะแนน)

1) Optimization Problem: หาเซตย่อยของ D ที่ให้ผลรวมเป็น k

Decision Problem:

2) Optimization Problem: หา tour ที่มีระยะทางสั้นสุดที่ผ่านทุกเมืองในกราฟ G

Decision Problem:

3) Optimization Problem: หาของที่หยิบใส่ถุงให้มีมูลค่ารวมสูงสุด โดยที่ถุงไม่ขาด

Decision Problem:

2. จากทฤษฎีการจัดกลุ่มความซับซ้อนของปัญหาด้านการคำนวณ จงเขียนชื่อเต็มและอธิบายความหมายโดยสังเขปของกลุ่มปัญหา (problem classes) ต่อไปนี้ *P, NP, co-NP, NPC, NP-Hard* และวาดแผนภาพเซ็ตแสดงกลุ่มปัญหาดังกล่าว บนสมมติฐานที่ว่า $P \neq NP$, $NP \neq co\text{-}NP$ (7 คะแนน)

/********** **จบตอนที่** 5 **********/