

FACULTY OF ENGINEERING
CHULALONGKORN UNIVERSITY
2110327 Algorithm Design

YEAR II, Second Semester, Final Examination, May 11, 2018, Time 13:00 – 16:00

ชื่อ-นามสกุล _____ เลขประจำตัว _____ CR58 _____

หมายเหตุ

1. ข้อสอบมีทั้งหมด 14 ข้อในกระดาษคำถามคำตอบ รวม จำนวน 8 หน้า คะแนนเต็ม 107 คะแนน
2. ไม่อนุญาตให้นำตำราและเอกสารใด ๆ เข้าในห้องสอบ
3. ห้ามการหยิบยืมสิ่งใดๆ ทั้งสิ้น จากผู้สอบอื่นๆ เว้นแต่เจ้าหน้าที่ควบคุมการสอบจะหยิบยืมให้
4. ห้ามนำส่วนใดส่วนหนึ่งของข้อสอบและสมุดคำตอบออกจากห้องสอบ
5. ผู้เข้าสอบสามารถออกจากห้องสอบได้ หลังจากผ่านการสอบไปแล้ว 45 นาที
6. เมื่อหมดเวลาสอบ ผู้เข้าสอบต้องหยุดการเขียนใดๆ ทั้งสิ้น
7. นิสิตกระทำผิดเกี่ยวกับการสอบ ตามข้อบังคับจุฬาลงกรณ์มหาวิทยาลัย มีโทษ คือ พ้นสภาพการเป็นนิสิต หรือ ได้รับ สัญลักษณ์ F ในรายวิชาที่กระทำผิด และอาจพิจารณาให้ถอนรายวิชาอื่นทั้งหมดที่ลงทะเบียนไว้ในภาคการศึกษานี้

ห้ามนิสิตพกโทรศัพท์หรืออุปกรณ์สื่อสารไว้กับตัวระหว่างสอบ หากตรวจพบจะถือว่านิสิตกระทำผิดเกี่ยวกับการสอบ อาจต้องพ้นสภาพการเป็นนิสิต หรือ ให้ได้รับ F และ อาจพิจารณาให้ถอนรายวิชาอื่นทั้งหมดที่ลงทะเบียนไว้ในภาคการศึกษานี้

*** ร่วมรณรงค์การกระทำผิด หรือทุจริตการสอบเป็นศูนย์ที่คณะวิศวกรรมศาสตร์ ***

ข้าพเจ้ายอมรับในข้อกำหนดที่กล่าวมานี้ ข้าพเจ้าเป็นผู้ทำข้อสอบนี้ด้วยตนเองโดยมิได้รับการช่วยเหลือ หรือให้ความช่วยเหลือในการทำข้อสอบนี้

ลงชื่อนิสิต

วันที่.....

ส่วนที่ 1: จงตอบคำถามต่อไปนี้

1. (12 คะแนน) จงตรวจสอบข้อความต่อไปนี้และทำเครื่องหมาย “✓” หน้าข้อความที่ถูกและทำเครื่องหมาย “X” หน้าข้อความที่ไม่ถูกต้อง (ตอบถูกได้ข้อละ 1 คะแนน ตอบผิดติดลบข้อละ 1 คะแนน ถ้าคะแนนรวมข้อ 1 ติดลบ จะคิดเป็น 0 คะแนน)

คำตอบ	ข้อความ
	1.1 ปัญหาใด ๆ ในกลุ่ม NPC นั้นสามารถ verify คำตอบ “Yes” ได้อย่างรวดเร็วเสมอ
	1.2 ปัญหาใด ๆ ในกลุ่ม NP-Hard นั้นสามารถ verify คำตอบ “Yes” ได้อย่างรวดเร็วเสมอ
	1.3 ถ้าปัญหา A ในกลุ่ม P ที่สามารถลดรูปไปเป็นปัญหา B ในกลุ่ม NP-Hard ได้ แสดงว่าปัญหา A นั้นต้องอยู่ในกลุ่ม NPC
	1.4 ไม่มีกราฟใดเลยที่ ลำดับของปมที่ค้นแบบ Depth First Search นั้นเหมือนกับลำดับของปมที่ค้นแบบ Breadth First Search
	1.5 อัลกอริทึม Depth First Search ไม่สามารถทำงานบน Directed Weighted Graph ได้
	1.6 อัลกอริทึมของ Dijkstra นั้นจะทำงานผิดพลาดเสมอ ถ้าหากกราฟมีเส้นเชื่อมที่มีน้ำหนักติดลบ
	1.7 สำหรับปัญหาการหาระยะทางสั้นสุดระหว่างทุก ๆ คู่ปม อัลกอริทึม Floyd-Warshall นั้นทำงานเร็วกว่าการใช้อัลกอริทึม Dijkstra เป็นจำนวน n ครั้งโดยแต่ละครั้งเริ่มจากแต่ละปม เสมอ
	1.8 ปัญหา SAT เป็นปัญหาในกลุ่ม NP-Hard
	1.9 ปัญหา Strongly Connected Component นั้นไม่อยู่ในกลุ่มปัญหา NP-C แล้วอย่างแน่นอน
	1.10 อัลกอริทึมของ Kruskal นั้นเป็นอัลกอริทึมแบบ Greedy
	1.11 ปัญหาการทอนเหรียญนั้นสามารถใช้วิธีการแก้ปัญหาแบบ Greedy มาแก้ได้อย่างถูกต้องเสมอ
	1.12 ปัญหาการตรวจสอบว่ากราฟนั้นมี Cycle หรือไม่ เป็นปัญหาในกลุ่ม P อย่างแน่นอน

2. (5 คะแนน) จากกราฟแบบมีทิศทางที่เก็บอยู่ในรูป adjacency list ดังข้างล่างนี้ จงระบุจำนวน strongly connected component ของกราฟดังกล่าวพร้อมด้วยระบุจำนวนปมในแต่ละ strongly connected component โดยให้เขียนเป็นรายการของตัวเลขจำนวนเต็ม เรียงจากน้อยไปมาก (ตัวอย่างเช่น สมมติว่าคำตอบคือมี 3 components แต่ละ component มีขนาดเป็น 3 ปม, 2 ปม และ 5 ปม ตามลำดับ ให้เขียนคำตอบเป็น 2, 3, 5)

0: 3, 4

คำตอบ: _____

1: 5

2: 3

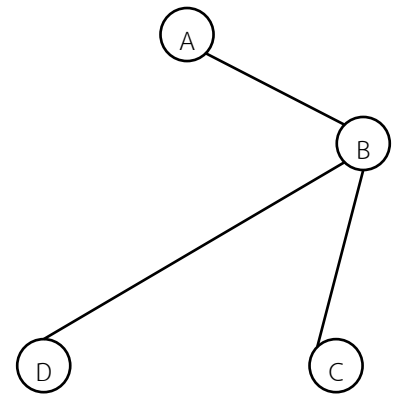
3: 1

4: 6

5: 1

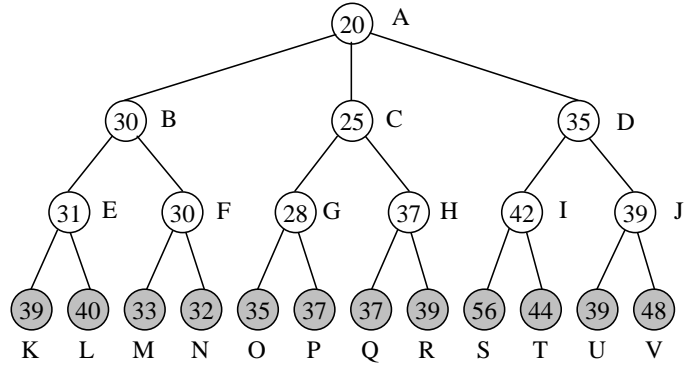
6: 0

3. (5 คะแนน) ปัญหา Independent Set เป็นดังนี้ กำหนดให้มีข้อมูลเข้าเป็นกราฟแบบไม่มีทิศทาง และค่า K เราต้องการทราบว่า มีเซตของปมในกราฟนั้นจำนวน K ปมหรือไม่ ที่แต่ละปมในเซตดังกล่าวไม่มีเส้นเชื่อมต่อกันเลย จากรูปกราฟด้านขวานี้ จงวาด State Space Tree เมื่อกำหนดค่า $k = 3$ ให้ระบุว่า state ที่ใช้นั้นคืออะไร เก็บในตัวแปรประเภทใด ในการวาด State Space Tree นั้นให้ระบุ state ในแต่ละปมด้วย การค้นให้ใช้วิธีแบบ Depth First Search โดยที่มีการใช้ Backtracking เป็น หยุดค้นต่อถ้า state นั้นมีคู่ปมที่มีเส้นเชื่อมระหว่างกัน



4. (5 คะแนน) จงวาดกราฟขนาด 4 ปม และให้ระบุปม S ในกราฟนั้น ที่ทำให้การทำอัลกอริทึม Depth First Search โดยเริ่มจาก S จะแวะผ่านปมทุกปมในลำดับเดียวกับที่อัลกอริทึม Breadth First Search จะผ่าน

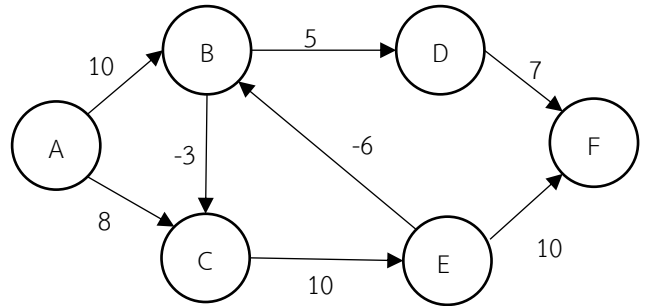
5. (5 คะแนน) รูปด้านล่างนี้แสดง State Space Tree ของปัญหาปัญหาหนึ่ง ซึ่งเราต้องการจะหาค่า solution state ที่มีค่าน้อยที่สุด กำหนดให้ปมที่เป็นใบเท่านั้นที่เป็น solution state และให้ค่าตัวเลขที่ระบุอยู่ในปมคือค่า cost function ของ state ดังกล่าว (สำหรับ solution state คือ cost จริง ส่วน partial state คือ lower bound ของ cost function) จงระบุปมที่เมื่อใช้วิธีการค้นแบบ Least Cost Search แล้ว ปมดังกล่าวไม่ถูกสร้างขึ้นมา โดยให้ระบุเป็นชื่อของปมที่เป็นตัวอักษรภาษาอังกฤษ เรียงจากน้อยไปมาก



ปมที่ไม่ถูกสร้างได้แก่: _____

6. (5 คะแนน) จงรัน Bellman-Ford Algorithm บนกราฟข้างล่างนี้โดยเริ่มจากปม A โดยให้ระบุค่าของ $D[a][b]$ ลงในตารางข้างล่าง เมื่อกำหนดให้ $D[a][b]$ คือ ระยะทางสั้นสุดในการเดินทางไปยังปม b โดยใช้เส้นเชื่อมไม่เกิน a เส้น

ค่า a	A	B	C	D	E	F
0	0	∞	∞	∞	∞	∞



7. (5 คะแนน) จงวาดกราฟขนาด 5 ปม แบบมีทิศทาง โดยที่กราฟดังกล่าวมี Topological Sorting ที่แตกต่างกันทั้งหมด $\binom{4}{2}$ แบบ

8. (5 คะแนน) จงเขียนผลลัพธ์ของการลดรูปปัญหา SAT ต่อไปนี้ให้เป็นปัญหา 3SAT

ปัญหา SAT $(\bar{x}_1 \vee \bar{x}_2)(x_1)(x_1 \vee x_2 \vee x_3 \vee \bar{x}_4)(x_2 \vee \bar{x}_3 \vee x_4)$

- ส่วนที่ 2: (ข้อละ 10 คะแนน) ในส่วนนี้จะเป็นการออกแบบอัลกอริทึม ในแต่ละข้อนิสิตต้องทำดังต่อไปนี้

- 1) บรรยายอัลกอริทึมเป็นคำอธิบาย (ไม่จำเป็นต้องเขียน source code) ซึ่งประกอบด้วยรายละเอียดมากเพียงพอที่จะเขียนโปรแกรมได้ นิสิตจะได้คะแนนตามความถูกต้อง ความชัดเจน และประสิทธิภาพของอัลกอริทึม
 - 2) สามารถเรียกใช้อัลกอริทึมหรือโครงสร้างข้อมูลที่ได้เคยเรียนมาในการอธิบายอัลกอริทึมได้โดยตรง ไม่จำเป็นต้องเขียน code แต่จะต้องระบุ input และ output ให้ชัดเจน (เช่น จะเรียกใช้ BFS ก็ต้องระบุให้ชัดเจนว่า input ของ BFS นั้นมีค่าเป็นอะไร สร้างมาอย่างไร และนำ output ของ BFS ไปใช้งานอย่างไรต่อ)
 - 3) **ต้องวิเคราะห์ประสิทธิภาพในการทำงานของอัลกอริทึมที่ออกแบบไว้ด้วย**
 - 4) **ถ้าพื้นที่เขียนไม่พอ ให้เขียนเพิ่มเติมในด้านหลังของ “หน้าที่ต้องเขียนคำตอบ” (อาจจะไม่ใช่หน้าเดียวกับตัวคำถาม)**
9. ปัญหา Meaningful Word Ladder คือการหาวิธีการเปลี่ยนคำภาษาอังกฤษหนึ่งคำที่มีความยาว n ตัวอักษรไปเป็นอีกคำหนึ่งที่มีความยาว n ตัวอักษรเช่นกันโดยการเปลี่ยนทีละตัวอักษร โดยที่ทุกๆคำระหว่างกลางจะต้องมีอยู่ใน dictionary เสมอโดยให้จำนวนครั้งที่ใช้เปลี่ยนน้อยที่สุดเท่าที่ทำได้ ตัวอย่างเช่น เริ่มจาก cat เปลี่ยนไปยัง dog สามารถเปลี่ยนได้ตามขั้นตอนดังนี้ $cat \rightarrow cot \rightarrow cog \rightarrow dog$ จงออกแบบอัลกอริทึมประเภท state space search ในการแก้ปัญหานี้ที่ทำงานได้เร็ว กำหนดให้ **st** และ **sf** เป็น ตัวแปรที่เก็บคำเริ่มต้น และ คำที่เป็นเป้าหมายตามลำดับ และให้ **dict** เป็นตัวแปรที่เก็บคำที่มีใน dictionary โดยสามารถเรียกใช้ **dict.has(x)** ซึ่งจะคืนค่า true ก็ต่อเมื่อมีคำ x ใน dictionary
- 1) ให้ระบุให้ชัดเจนว่า state คืออะไร เก็บในตัวแปรแบบไหน มี initial state เป็นอะไร และจาก partial state แต่ละอันจะผลิต state ใหม่อะไรบ้าง

- 2) จะใช้รูปแบบการค้นหาแบบไหน มี backtracking หรือ branch & bound หรือไม่ ถ้ามี เป็นอย่างไร และจะหยุดค้นเมื่อไร

3) ระบุเวลาการทำงานทั้งหมดรวมถึงเวลาที่ใช้ในการทำ backtracking หรือ branch & bound ด้วย

10. สมชายต้องการขับรถจากกรุงเทพฯ ไปยังเชียงใหม่ โดยรถของสมชายจะวิ่งได้ระยะทาง k กิโลเมตรหากน้ำมันเต็มถัง โดยสมชายได้กำหนดเส้นทางที่จะใช้เดินทางไว้แล้วและในระหว่างทางมีปั้มน้ำมันที่ระยะ $d_1 < d_2 < d_3 < \dots < d_n$ โดยที่ d_i คือระยะทางจากกรุงเทพฯ ถึงปั้มน้ำมันที่ i และ d_n คือปั้มน้ำมันที่อยู่เชียงใหม่ (เป็นจุดหมายของการเดินทาง) เพื่อประหยัดเวลาในการเดินทาง สมชายอยากจะทำให้น้ำมันน้อยครั้งที่สุดเท่าที่เป็นไปได้โดยที่น้ำมันต้องไม่หมดระหว่างทาง จงออกแบบอัลกอริทึมเพื่อคำนวณจำนวนปั้มที่ต้องแวะที่น้อยที่สุด จากข้อมูล n, k และ d_i ที่ให้ รับประกันว่ามีวิธีเติมน้ำมันที่ทำให้น้ำมันไม่หมดระหว่างการเดินทางอย่างน้อยหนึ่งวิธี

11. ประธานาธิบดีแห่งประเทศสาร์ซันท์กำลังจะมาเยือนเมืองสมุทรบุรี เพื่อเป็นการป้องกันการลอบสังหาร ตำรวจได้ติดตั้งอุปกรณ์ระบุตำแหน่งจากเสียงปืน อุปกรณ์นี้เป็นเสาซึ่งจะรับเสียงปืน เมื่อได้รับเสียงปืนแล้วก็จะทราบระยะห่างของเสากับปืนที่ยิง เพื่อให้สามารถระบุตำแหน่งได้ เราจะติดตั้งเสานี้หลาย ๆ เสา ถ้าหากคนที่ยิงปืน อยู่ภายในพื้นที่สามเหลี่ยมที่มีจุดยอดเป็นเสาดังกล่าว กำหนดให้เรามีเสาอยู่ n ต้นคือ v_1, \dots, v_n โดยที่ $p[i].x$ และ $p[i].y$ เป็นตัวแปรที่ระบุตำแหน่ง ในพื้นที่ 2 มิติ ของเสา v_i เราต้องการทราบว่า ถ้ามีการยิงปืนในพิกัด (x, y) แล้ว เราจะสามารถระบุตำแหน่งของปืนนั้นได้หรือไม่ จงเขียนรหัสเทียมของวิธีการโดยสังเขปว่าเราสามารถแก้ปัญหานี้ได้อย่างไร พร้อมทั้งระบุประสิทธิภาพเชิงเวลาด้วย

```
bool can_detect_gun_shot(p, x, y)
```

12. กำหนดให้มีตาราง 2 มิติขนาด r แถว c คอลัมน์อยู่ แต่ละแถวกำกับด้วยหมายเลข 1 ถึง r และแต่ละคอลัมน์กำกับด้วยหมายเลข 1 ถึง c เราอยู่ ณ ช่อง r_1, c_1 และต้องการเดินไปยังช่อง r_2, c_2 โดย เมื่อเราอยู่ ณ ช่อง a, b ใด ๆ ก็ตาม เราสามารถเดินทางไปยังช่อง 8 ช่องใดก็ได้ที่อยู่รอบตัว และไม่อยู่นอกตาราง การเดินผ่านช่อง a, b ใด ๆ นั้นจะต้องเสียเงินเป็นจำนวน $p[a][b]$ จงออกแบบอัลกอริทึมสำหรับคำนวณจำนวนเงินที่น้อยที่สุดที่ใช้ในการเดินทางจากช่อง r_1, c_1 ไปยังช่อง r_2, c_2 จากข้อมูล r, c, r_1, c_1, r_2, c_2 และ p

13. หมู่บ้านแห่งหนึ่งมีคนอยู่ n คน ให้คนแต่ละคนระบุได้ด้วยตัวเลข 1 ถึง n คนในหมู่บ้านนี้บางคนจะเป็นเพื่อนกัน ให้ bool `friend[a][b]` เป็นตัวแปรที่เก็บข้อมูลความเป็นเพื่อน โดย `friend[a][b]` จะมีค่าเป็นจริงก็ต่อเมื่อ a และ b เป็นเพื่อนกัน เรานิยามคำว่า “เพื่อนของเพื่อน” ดังนี้ a จะเป็น “เพื่อนของเพื่อน” กับ b ก็ต่อเมื่อมีคน c_1, \dots, c_k โดยที่ a เป็นเพื่อนกับ c_1 , c_1 เป็นเพื่อนกับ c_2 , ..., $c_{(k-1)}$ เป็นเพื่อนกับ c_k โดยที่ c_k ก็เป็นเพื่อนกับ b นอกจากนี้ คนที่เป็นเพื่อนกันก็ถือว่าเป็น “เพื่อนของเพื่อน” ด้วยเช่นเดียวกัน
- มันเป็นไปได้ที่จะมีคู่คนบางคนในหมู่บ้านนี้ไม่ได้เป็น “เพื่อนของเพื่อน” กัน จงออกแบบอัลกอริทึมที่คำนวณว่า ถ้าเราต้องการให้คนทุกคนในหมู่บ้านนี้เป็นเพื่อนของเพื่อนกันทั้งหมด เราจะต้องทำให้คู่คนกี่คู่เป็นเพื่อนกัน โดยให้อัลกอริทึมนี้รับข้อมูลนำเข้าเป็น n และ `friend` และตอบค่าจำนวนเต็มกลับมา (ถ้าหากคนทั้งหมู่บ้านเป็นเพื่อนของเพื่อนกันอยู่ทั้งหมดแล้ว ให้ตอบ 0)

14. ฝนตกหนักมากในเมืองสมุทรบุรี ทำให้ถนนหลายสายไม่สามารถใช้งานได้ เมืองสมุทรบุรีมีอำเภอทั้งหมด n อำเภอ กำกับด้วยหมายเลข 1 ถึง n ก่อนที่ฝนจะตกนั้น ทุกอำเภอสามารถเดินทางไปหากันได้ด้วยโครงข่ายถนนสองทิศทางที่เชื่อมอำเภอต่าง ๆ เข้าด้วยกัน ให้ $\text{bool } e[a][b]$ เป็นตัวแปรที่จะมีค่าเป็นจริงก็ต่อเมื่อ มีถนนเชื่อมต่อระหว่างอำเภอ a กับอำเภอ b และให้ $\text{bool } f[a][b]$ เป็นตัวแปรที่จะเป็นจริงก็ต่อเมื่อ ถนนที่เชื่อมระหว่างอำเภอ a และ b นั้นโดนน้ำท่วมจนขาดออกจากกัน เราต้องการให้การเดินทางระหว่างอำเภอต่าง ๆ สามารถทำได้เช่นเดิม โดยเราจะทำการซ่อมแซมถนนบางเส้นให้กลับมาใช้งานได้ กำหนดให้ $\text{int } c[a][b]$ เป็นตัวแปรที่ระบุค่าใช้จ่ายในการซ่อมถนนซึ่งเชื่อมระหว่างเมือง a และ b จงออกแบบอัลกอริทึมที่หาจำนวนงบประมาณน้อยที่สุดที่ซ่อมแซมถนนให้ทุกเมืองสามารถเดินทางไปมาถึงกันได้ผ่านถนนเส้นต่าง ๆ จากข้อมูล n, e, f, c ที่กำหนดให้ (การเดินทางไปมาหากันได้นั้นหมายถึงเราสามารถเดินทางจากเมืองหนึ่งผ่านถนนที่ใช้งานได้ไปยังเมืองต่าง ๆ และจบที่เมืองปลายทาง ไม่จำเป็นต้องมีถนนเชื่อมต่อระหว่างทุกคูเมืองก็เป็นได้ ตัวอย่างเช่น ถ้ามีสามอำเภอ และมีถนนเชื่อมระหว่าง 1 กับ 2 และ ระหว่าง 2 กับ 3 ก็ถือว่าอำเภอ 1 สามารถเดินทางไปยังอำเภอ 3 ได้เช่นกัน)