

FACULTY OF ENGINEERING  
CHULALONGKORN UNIVERSITY  
2110327 Algorithm Design

YEAR III, First Semester, Final-term Examination, November 28, 2016, Time 8:30 – 11:30

ชื่อ-นามสกุล \_\_\_\_\_ เลขประจำตัว \_\_\_\_\_ CR58 \_\_\_\_\_

หมายเหตุ

1. ข้อสอบมีทั้งหมด 10 ข้อในกระดาษคำถามคำตอบ รวม จำนวน 7 หน้า คะแนนเต็ม 90 คะแนน
2. ไม่อนุญาตให้นำตำราและเครื่องคำนวณต่างๆ ใดๆ เข้าห้องสอบ
3. ควรเขียนตอบด้วยลายมือที่อ่านง่ายและชัดเจน
4. ห้ามการหยิบยืมสิ่งใดๆ ทั้งสิ้น จากผู้สอบอื่นๆ เว้นแต่ผู้คุมสอบจะหยิบยืมให้
5. ห้ามนำส่วนใดส่วนหนึ่งของข้อสอบออกจากห้องสอบ ข้อสอบเป็นทรัพย์สินของราชการซึ่งผู้ลักพาอาจมีโทษทางคดีอาญา
6. ผู้เข้าสอบสามารถออกจากห้องสอบได้ หลังจากผ่านการสอบไปแล้ว 45 นาที
7. เมื่อหมดเวลาสอบ ผู้เข้าสอบต้องหยุดการเขียนใดๆ ทั้งสิ้น
8. นิสิตกระทำผิดเกี่ยวกับการสอบ ตามข้อบังคับจุฬาลงกรณ์มหาวิทยาลัย มีโทษ คือ ได้รับ สัญลักษณ์ F ในรายวิชาที่ทุจริต และพักการศึกษาอย่างน้อย 1 ภาคการศึกษา

**ห้านิสิตพบโทรศัพท์หรืออุปกรณ์สื่อสารไว้กับตัวระหว่างสอบ หากตรวจพบจะถือว่านิสิตกระทำผิดเกี่ยวกับการสอบ ให้ได้รับ F และอาจพิจารณาสั่งพักการศึกษา \*\***

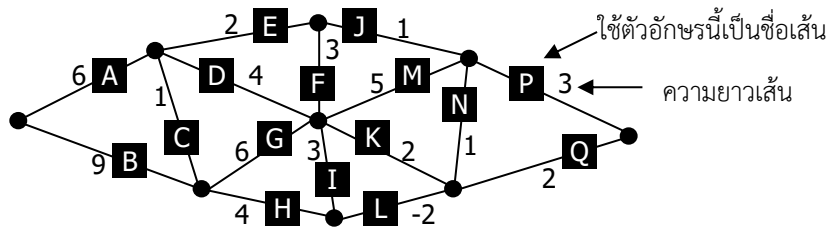
ข้าพเจ้ายอมรับในข้อกำหนดที่กล่าวมานี้ ข้าพเจ้าเป็นผู้ทำข้อสอบนี้ด้วยตนเองโดยมิได้รับการช่วยเหลือ หรือให้ความช่วยเหลือในการทำข้อสอบนี้

ลงชื่อนิสิต .....

วันที่.....

1. (20 คะแนน) จงระบุว่า ข้อย่อต่อไปนี้ข้อใดถูก ข้อใดผิด (ไม่ต้องอธิบายที่มาของคำตอบ เขียนตอบแค่ถูกหรือผิด)  
ข้อใดที่เป็นคำถามเกี่ยวกับกราฟ ให้  $v$  คือจำนวนปม และ  $e$  คือจำนวนเส้นเชื่อมของกราฟ
  - 1) ..... ถ้าการทำ shortest path จากปม  $s$  ไปยังปม  $t$  ในกราฟ  $G$  ใช้เวลา  $\Theta(f(v,e))$  แสดงว่า เราต้องใช้เวลา  $\Omega(v f(v, e))$  เพื่อหา shortest path จาก  $s$  ไปยังทุก ๆ ปมในกราฟ  $G$
  - 2) ..... การทำ depth first search ในกราฟ  $G$  ที่แทนด้วย adjacency matrix จะใช้เวลาแค่  $O(v)$
  - 3) ..... เราหา strongly connected components ต่าง ๆ ใน directed graph  $G$  ในเวลาแค่  $O(v + e)$
  - 4) ..... เราทดสอบว่า directed graph  $G$  เป็น directed acyclic graph หรือไม่ ได้ในเวลาแค่  $O(v + e)$
  - 5) ..... เราหา shortest path จากปม  $s$  ไปยัง  $t$  ในกราฟ  $G$  ที่เส้นเชื่อมทุกเส้นยาวเท่ากันหมดได้ในเวลาแค่  $O(v + e)$
  - 6) ..... เราหา minimum spanning tree ในกราฟ  $G$  ที่เส้นเชื่อมทุกเส้นยาวเท่ากันหมดได้ในเวลาแค่  $O(v + e)$
  - 7) ..... การใช้ Dijkstra's algorithm เพื่อหา shortest path ในกราฟ  $G$  กราฟหนึ่งที่มีความยาวเส้นเชื่อมบางเส้นเป็นลบ ก็อาจได้คำตอบที่ถูกต้องก็ได้
  - 8) ..... ให้  $W$  คือ adjacency matrix ของกราฟ  $G$  และให้  $D=(W)^v$  จะได้ว่า  $D[a][b]$  คือระยะทางของ shortest path จากปม  $a$  ไป  $b$
  - 9) ..... ถ้าเขียนอัลกอริทึมของ Floyd-Warshall แบบตรงไปตรงมา จะใช้เวลาทำงานเป็น  $\Theta(v^4)$
  - 10) ..... ให้  $k$  คือความยาวของเส้นเชื่อมที่สั้นสุดในกราฟ  $G$  ถ้าเราลบความยาวของเส้นเชื่อมทุกเส้นลง  $k$  จะไม่ทำให้ shortest path ระหว่างปมต่าง ๆ ใน  $G$  เปลี่ยนแปลง

- 11) ..... ให้เส้นเชื่อมของกราฟ G มี capacity เท่ากับ 1 เท่ากันหมดทุกเส้น หลังจากเราใช้ Ford-Fulkerson เพื่อหา max flow ใน G จะได้ว่า max flow ของ G มีค่าเท่ากับจำนวนเส้นเชื่อมที่พุ่งออกจากปม source ของ G
  - 12) ..... ให้เส้นเชื่อมของกราฟ G มี capacity เท่ากับ 1 เท่ากันหมดทุกเส้น หลังจากเราใช้ Ford-Fulkerson เพื่อหา max flow ใน G จะได้ว่า min cut ของ G มีค่าเท่ากับจำนวนเส้นเชื่อมที่พุ่งเข้าหาจากปม sink ของ G
  - 13) ..... การค้นคำตอบใน state space แบบ depth-first search, breadth-first search และ least-cost search อาศัยโครงสร้างข้อมูลแบบ stack, queue, และ priority queue ตามลำดับ ในการเก็บ states ต่าง ๆ ระหว่างการค้น
  - 14) ..... ถ้าให้ cost สุ่มๆ กับ state ต่าง ๆ ระหว่างการทำ least-cost search เราอาจไม่พบคำตอบก็ได้ (ถึงแม้ว่าจะมี answer state อยู่ก็ตาม)
  - 15) ..... เราต้องมี function ในการหา lower bound ของ cost ประจำ state ระหว่างการค้นคำตอบด้วยวิธี branch and bound เพื่อแก้ปัญหาประเภท minimization problem
  - 16) ..... หากเรามี lower bound function อยู่หลายฟังก์ชันที่สามารถใช้ระหว่างการค้นคำตอบด้วยวิธี branch and bound ได้ เราควรเลือก lower bound function ตัวที่ให้ค่าน้อยสุดๆ ยิ่งน้อยยิ่งดี
  - 17) ..... ปัญหา graph isomorphism เป็นหนึ่งในปัญหา NP และยังไม่มีการหา polynomial-time algorithm เพื่อหาคำตอบให้กับปัญหานี้ได้ นี่แสดงว่า ปัญหานี้เป็นปัญหาในกลุ่ม NP-Complete
  - 18) ..... การจะพิสูจน์ว่า  $P = NP$  ก็เพียงแค่หา polynomial-time algorithm ให้กับเพียงหนึ่งปัญหาในกลุ่ม NP-Complete ก็พิสูจน์ได้แล้ว
  - 19) ..... ปัญหา 1SAT, 2SAT, 3SAT เป็นปัญหาในกลุ่ม NP ที่มีความยากง่ายเท่ากันหมด
  - 20) ..... สมชายเสนออัลกอริทึมแบบ brute force เพื่อหา Longest common subsequence ของ X และ Y ด้วยการแจกแจงทุก subsequences ของ X และ Y เพื่อหาว่าคู่ใดเหมือนกันและยาวสุด ซึ่งใช้เวลา  $O(2^n 2^m)$  โดยที่ X และ Y มีความยาว n และ m ตามลำดับ นั้นย่อมแสดงว่า อัลกอริทึมที่สมชายเสนอมายจัดอยู่ในกลุ่ม NP-complete
2. (3 คะแนน) จงเขียนลำดับของชื่อเส้นเชื่อมที่ถูกเลือกให้เป็นส่วนหนึ่งของ minimum spanning tree ด้วยการ ใช้ Prim's algorithm กับกราฟข้างล่างนี้ โดยเริ่มที่ปมซ้ายสุดในรูป หาก ณ ขณะใดระหว่างการหา สามารถเลือกได้หลายเส้นที่ให้ผลเหมือนกัน ก็ให้เลือกเส้นที่มีชื่อตัวอักษร "ตัวน้อยกว่า" (ไม่ต้องแสดงวิธีทำ)



- 3. (3 คะแนน) หลังจากที่เราใช้อัลกอริทึมของ Floyd Warshall เพื่อหาความยาวของ shortest paths ในกราฟ G แล้ว จงบรรยายวิธีที่เร็วที่สุด ๆ เพื่อหาว่า "มี negative-weight cycle ใน G หรือไม่?"

.....

.....

4. (4 คะแนน) ปัญหา graph coloring ถามว่า ถ้าเรามีสี  $k$  สี เราจะให้สีกับปมต่าง ๆ ในกราฟ  $G$  ได้หรือไม่ โดยที่ปมปลายของเส้นเชื่อมเดียวกันต้องมีสีต่างกัน ปัญหานี้แก้ได้ด้วยอัลกอริทึมข้างล่างนี้

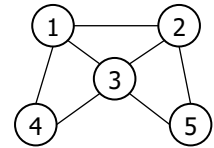
```

boolean is_k_colorable(G, k) {
    n = number_of_vertices(G)
    return is_k_colorable(G, k, new int[n], 0)
}
boolean is_k_colorable(G, k, c[1..n], m) {
    if (n == m) {
        return true
    } else {
        for (color = 1 to k) {
            c[m+1] = color // ลองให้สีปมที่ m+1 ด้วยสี color
            if valid_coloring(G, c, m+1) // ตรวจสอบว่าการให้สีปมใน c จาก c[1] ถึง c[m+1] ถูกตามกฎหรือไม่
                if is_k_colorable(G, k, c, m+1) return true
        }
    }
    return false
}

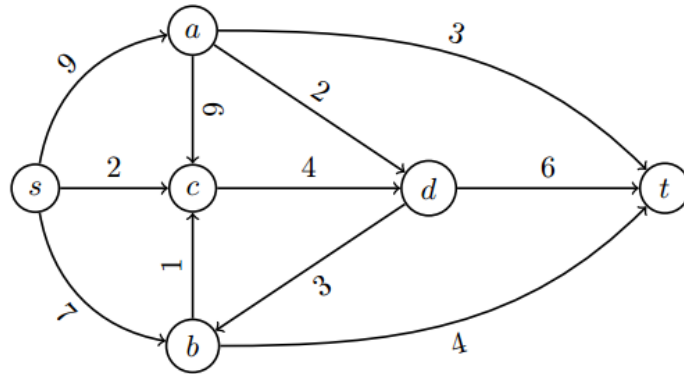
```

หากใช้อัลกอริทึมข้างบนนี้ทำงานกับกราฟทางขวานี้ โดยที่  $k = 3$

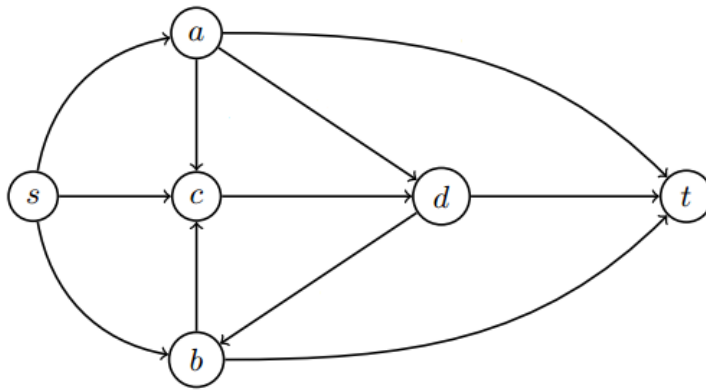
- จงวาด state ที่เกิดขึ้นระหว่างการค้นคำตอบ และเขียนเส้นแสดงความสัมพันธ์ระหว่าง state เหล่านี้ด้วยว่า state ใดเกิดมาจาก state ใดบ้างจนพบคำตอบ
- และ อยากทราบว่า อัลกอริทึมข้างบนนี้ใช้กลวิธีการค้นคำตอบที่เรียกว่าอะไร



5. (10 คะแนน) จากกราฟข้างล่างนี้ที่แต่ละเส้นมีความจุระบุอยู่



a) (5 คะแนน) จงหา max flow ของกราฟนี้ โดยให้ระบุปริมาณที่ไหลในแต่ละเส้น กำหนดให้ s คือ source และ t คือ sink



b) (3 คะแนน) จงระบุน้ำหนักของ edge ดังต่อไปนี้ใน residue กราฟตอนที่เป็น max flow

- a.  $s \rightarrow a$  \_\_\_\_\_
- b.  $a \rightarrow s$  \_\_\_\_\_
- c.  $c \rightarrow d$  \_\_\_\_\_
- d.  $d \rightarrow c$  \_\_\_\_\_
- e.  $d \rightarrow t$  \_\_\_\_\_

c) (2 คะแนน) จงระบุ Min-Cut ของกราฟนี้ โดยให้ระบุเซตของเส้นเชื่อมที่เป็น Min-Cut

6. (10 คะแนน) ปัญหา Equal Subset ให้  $X = \{x_1, x_2, \dots, x_n\}$  เป็นเซตของเลขจำนวนจริง ในข้อนี้เราอยากหาว่ามี subset A ใดของ X ไหมที่  $\sum_{x \in A} x = \sum_{y \in X-A} y$  จงพิสูจน์ว่า Equal Subset อยู่ใน NP-Hard (นิสิตสามารถอ้างอิงถึงปัญหา NP-Complete อื่นๆที่แสดงในท้องได้โดยไม่ต้องพิสูจน์)

ข้อ 7 – 11 เป็นการออกแบบอัลกอริทึม ในทุก ๆ ข้อนี้จะต้องทำดังต่อไปนี้

- 1) บรรยายอัลกอริทึมเป็นคำอธิบาย (ไม่จำเป็นต้องเขียน source code) ซึ่งประกอบด้วยรายละเอียดมากเพียงพอที่จะเขียนโปรแกรมได้ นิสิตจะได้คะแนนตามความถูกต้อง ความชัดเจน และประสิทธิภาพของอัลกอริทึม
  - 2) นิสิตสามารถใช้อัลกอริทึมหรือโครงสร้างข้อมูลที่ได้เคยเรียนมาในการอธิบายอัลกอริทึมได้โดยตรง ไม่จำเป็นต้องเขียน code
  - 3) วิเคราะห์ประสิทธิภาพในการทำงานของอัลกอริทึมที่ออกแบบไว้
7. (10 คะแนน) ปัญหา MAX-3SAT เป็นดังนี้ ข้อมูลนำเข้าคือ Boolean Expression ในรูป conjunctive normal form เช่น

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

โดยมีตัวแปร Boolean  $x_1, \dots, x_n$  อยู่  $n$  ตัวและมี clause (หนึ่งชุดของวงเล็บ) ที่  $\wedge$  (and) กันอยู่  $m$  พจน์โดยในแต่ละพจน์ประกอบด้วย ตัวแปรในรูป positive หรือ negative form สามตัวที่  $\vee$  (or) กันอยู่ โดยปัญหาในข้อนี้คือหาว่า  $x_1, \dots, x_n$  แต่ละตัวจะต้องเป็น false หรือ true ที่จะให้มีจำนวนพจน์มากที่สุดที่ true เท่าที่ทำได้ จงอธิบายวิธีแก้ปัญหานี้ด้วย State-Space Search และ Branch and Bound

8. (10 คะแนน) กำหนดให้  $D$  เป็น adjacency matrix ของกราฟแบบมีทิศทาง  $G$  ขนาด  $n$  ปม โดยที่  $D[a][b]$  คือน้ำหนักของเส้นเชื่อมจากปม  $a$  ไป  $b$  เราต้องการทราบค่าระยะทางสั้นสุดจากปม  $s$  ไปยังปมใด ๆ ในกราฟ  $D$  นี้ โดยที่มีข้อกำหนดเพิ่มเติมคือระยะทางสั้นสุดนั้นจะต้องเกิดจาก path ที่มีเส้นเชื่อมจำนวน  $K$  เส้นเชื่อม (ซ้ำกันได้) พอดี ตัวอย่างเช่น path  $\langle 1, 3, 1, 3, 5 \rangle$  จะถือว่าเป็น path ที่มี 4 เส้นเชื่อมถึงแม้ว่าจะวิ่งผ่านเส้นเชื่อม  $(1,3)$  จำนวน 2 ครั้งก็ตาม จงออกแบบอัลกอริทึมสำหรับการแก้ปัญหานี้ เมื่อกำหนดให้ข้อมูลนำเข้าคือ  $D, K, s$  และ  $n$
9. (10 คะแนน) ในจังหวัดที่ห่างไกลจังหวัดหนึ่งมีถนนอยู่  $m$  เส้น ซึ่งเชื่อมตำบล  $n$  ตำบลเข้าด้วยกัน ทุกตำบลสามารถเดินทางไปหากันได้โดยใช้ถนนเหล่านี้ ในเวลากลางคืน เราจะต้องเปิดไฟบนถนนเพื่อความปลอดภัย กำหนดให้  $D[a][b]$  คือกำลังไฟฟ้าที่ต้องใช้ในการเปิดไฟบนถนนที่เชื่อมตำบล  $a$  กับตำบล  $b$  มีหน่วยเป็นวัตต์ อย่างไรก็ตาม การไฟฟ้าของจังหวัดจะต้องเดินสายไฟเพื่อจ่ายไฟให้กับถนน การไฟฟ้าเลือกที่จะใช้สายไฟที่รองรับการจ่ายไฟได้ไม่เกิน  $K$  วัตต์ ทำให้ถนนเส้นที่ใช้ไฟมากกว่านั้นไม่มีไฟส่องสว่าง เราจะกำหนดให้ทุก ๆ ตำบลที่สามารถเดินทางไปหากันได้ตอนกลางคืนผ่านถนนที่มีไฟส่องสว่างเปิดอยู่ นั้นเป็นตำบลในอำเภอเดียวกัน ส่วนตำบลที่เดินทางไปหากันไม่ได้นั้นอยู่คนละอำเภอ จงออกแบบอัลกอริทึมเพื่อหาค่า  $K$  ที่น้อยที่สุดที่ทำให้จังหวัดนี้มีจำนวนอำเภอไม่เกิน  $A$  อำเภอ เมื่อกำหนดให้ข้อมูลนำเข้าคือ  $D, A, m$  และ  $n$

10. (10 คะแนน) บริษัทแห่งหนึ่งมีพนักงาน  $n$  คน สำหรับคู่พนักงาน  $(a,b)$  ใด ๆ เราทราบว่าพนักงาน  $a$  นั้น “นับถือ” พนักงาน  $b$  หรือไม่ ให้  $R[a][b]$  มีค่าเป็นจริงก็ต่อเมื่อ พนักงาน  $a$  นับถือพนักงาน  $b$  ให้สังเกตว่าถ้า  $a$  นับถือ  $b$  นั้นไม่ได้จำเป็นที่  $b$  จะต้องนับถือ  $a$  ด้วย และกำหนดให้ว่า ถ้า  $a$  นับถือ  $b$  และ  $b$  นับถือ  $c$  เราจะถือว่า  $a$  นับถือ  $c$  โดยอ้อม นอกจากนี้  $a$  จะนับถือโดยอ้อมต่อทุก ๆ คนที่  $a$  นับถือ โดยอ้อมนั้นนับถือ เจ้าของบริษัทต้องการเลือกหัวหน้าพนักงาน โดยมีเงื่อนไขคือหัวหน้าพนักงานนั้นจะต้องได้รับการนับถือไม่ว่าจะทางตรงหรือทางอ้อม จากทุก ๆ คนในบริษัท จงออกแบบอัลกอริทึมสำหรับหาคนที่สามารถทำหน้าที่เป็นหัวหน้าพนักงานได้ เมื่อกำหนดให้ข้อมูลนำเข้าคือ  $R$ , และ  $n$