

FACULTY OF ENGINEERING
CHULALONGKORN UNIVERSITY
2110327 Algorithm Design

YEAR III, First Semester, Final Examination, December 4, 2017, Time 8:30 – 11:30

ชื่อ-นามสกุล _____ เลขประจำตัว _____ CR58 _____

หมายเหตุ

1. ข้อสอบมีทั้งหมด 12 ข้อในกระดาษคำถามคำตอบ รวม จำนวน 9 หน้า คะแนนเต็ม 115 คะแนน
2. ไม่อนุญาตให้นำตำราและเอกสารใด ๆ เข้าในห้องสอบ
3. ห้ามการหยิบยืมสิ่งใดๆ ทั้งสิ้น จากผู้สอบอื่นๆ เว้นแต่ผู้คุมสอบจะหยิบยืมให้
4. ห้ามนำส่วนใดส่วนหนึ่งของข้อสอบและสมุดคำตอบออกจากห้องสอบ
5. ผู้เข้าสอบสามารถออกจากห้องสอบได้ หลังจากผ่านการสอบไปแล้ว 45 นาที
6. เมื่อหมดเวลาสอบ ผู้เข้าสอบต้องหยุดการเขียนใดๆ ทั้งสิ้น
7. นิติกรจะทำผิดเกี่ยวกับการสอบ ตามข้อบังคับจุฬาลงกรณ์มหาวิทยาลัย มีโทษ คือ พ้นสภาพการเป็นนิสิต หรือ ได้รับ สัญลักษณ์ F ในรายวิชาที่กระทำผิด และอาจพิจารณาให้ถอนรายวิชาอื่นทั้งหมดที่ลงทะเบียนไว้ในภาคการศึกษานี้

ห้ามนิสิตพกโทรศัพท์หรืออุปกรณ์สื่อสารไว้กับตัวระหว่างสอบ หากตรวจพบจะถือว่านิสิตกระทำผิดเกี่ยวกับการสอบ อาจต้องพ้นสภาพการเป็นนิสิต หรือ ให้ได้รับ F และ อาจพิจารณาให้ถอนรายวิชาอื่นทั้งหมดที่ลงทะเบียนไว้ในภาคการศึกษานี้

*** ร่วมรณรงค์การกระทำผิด หรือทุจริตการสอบเป็นศูนย์ที่คณะวิศวกรรมศาสตร์ ***

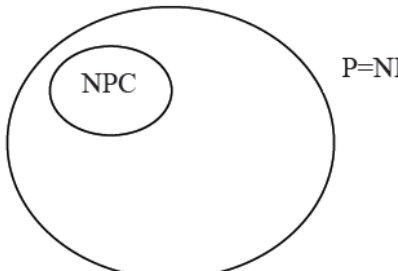
ข้าพเจ้ายอมรับในข้อกำหนดที่กล่าวมานี้ ข้าพเจ้าเป็นผู้ทำข้อสอบนี้ด้วยตนเองโดยมิได้รับการช่วยเหลือ หรือให้ความช่วยเหลือในการทำข้อสอบนี้

ลงชื่อนิสิต

วันที่.....

1. (15 คะแนน) จงตรวจสอบข้อความต่อไปนี้และทำเครื่องหมาย “✓” หน้าข้อความที่ถูกและทำเครื่องหมาย “X” หน้าข้อความที่ไม่ถูกต้อง

	The problem of determining whether there exists a cycle in an undirected graph is in NP.
	If a new problem is in NP and we can reduce a known NP-Complete Problem to it, then the new problem is NP -Complete
	For all problems $X \in NP$, X can be reduced to 3-Coloring in polynomial time.
	If Vertex-Cover $\in P$ then Sat $\in P$.
	If $P = NP$ then Shortest-Path is NP-complete.
	It is possible that Independent-Set $\in P$ and Ham-cycle $\notin P$
	Let X_1 and X_2 be decision problems in NP, and assume $P \neq NP$. If X_1 can be reduced to X_2 in polynomial time and X_2 can be reduced to X_1 in polynomial time, then both X_1 and X_2 are NP-complete.
	Let S be an NP-complete problem and Q and R be two other problems not known to be in NP. Q is polynomial time reducible to S and S is polynomial-time reducible to R . R is NP-hard.
	Assuming $P \neq NP$, then $NP\text{-complete} \cap P = \emptyset$
	Let X be a problem that belongs to the class NP. If X can be solved deterministically in polynomial time, then $P = NP$.
	Suppose a polynomial time algorithm is discovered that correctly computes the largest clique in a given graph. In this scenario, the following represents the correct Venn diagram of the

	<p>complexity classes P, NP and NP Complete (NPC).</p>  <p>The diagram shows a large circle labeled 'P=NP' containing a smaller circle labeled 'NPC', indicating that NPC is a subset of P=NP.</p>
	<p>Given a graph where all edge lengths are integers (possibly negative), there is a polynomial-time algorithm to test whether the graph contains a negative cycle (i.e., a cycle where the sum of the lengths of the edges in the cycle is negative).</p>
	<p>The search problem “Given a graph G with n vertices, find a way to color the graph with n – 1 colors, or report that no such coloring exists” is in P.</p>
	<p>In SUBSET-SUM the input is a set $S = \{x_1, x_2, \dots, x_n\}$ and the integer t. In the COMPOSITE problem you are given as input an integer y and the question to answer is whether or not y has a factor besides 1 and itself. If SUBSET-SUM is NP-complete and COMPOSITE is in NP, then SUBSET-SUM can be reduced to COMPOSITE in polynomial time.</p>
	<p>If there is an $O(n\sqrt{t})$ algorithm for SUBSET-SUM, then $P = NP$</p>

2. (10 คะแนน) จากกราฟไม่ระบุทิศทางดังด้านล่างนี้ที่เก็บอยู่ในรูป adjacency matrix กำหนดให้ปมในกราฟนี้กำกับด้วยหมายเลข 1 ถึง 7 โดยให้คอลัมน์แรกระบุถึงปมหมายเลข 1 และคอลัมน์อื่น ๆ เรียงตามลำดับ

1	1	0	0	0	1	0
1	1	1	0	0	0	0
0	1	1	0	0	1	1
0	0	0	1	1	0	0
0	0	0	1	1	1	0
1	0	1	0	1	1	0
0	0	1	0	0	0	1

กำหนดให้ DFS Tree คือต้นไม้ที่ระบุปมต่าง ๆ ที่ อัลกอริทึม DFS แวะผ่าน โดยจะมีเส้นเชื่อมจากปม x ไป y ก็ต่อเมื่อ DFS ทำการแวะผ่านปม y โดยใช้เส้นเชื่อมจาก x ไปยัง y

BFS Tree ก็นิยามเช่นเดียวกัน โดยใช้ อัลกอริทึม BFS แทน

- a) (5 คะแนน) จงวาด Depth First Search Tree ที่เกิดจากการทำ DFS เริ่มที่ปม 1

- b) (5 คะแนน) จงวาด Breadth First Search Tree ที่เกิดจากการทำ BFS เริ่มที่ปม 1

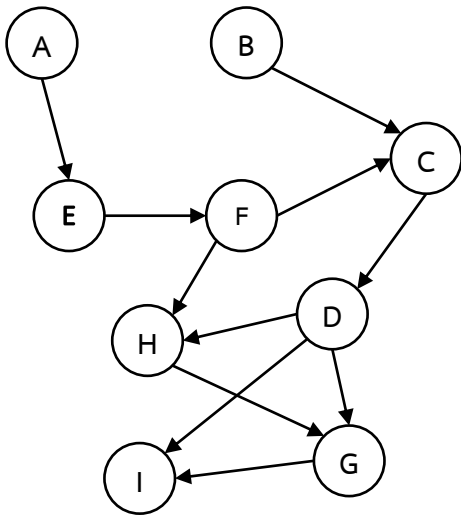
3. (10 คะแนน) จากกราฟไม่ระบุทิศทางมีน้ำหนักด้านล่างนี้ที่ระบุด้วยที่เก็บอยู่ในรูป adjacency matrix กำหนดให้ปมในกราฟนี้กำกับด้วยหมายเลข 1 ถึง 7 โดยให้คอลัมน์แรกระบุถึงปมหมายเลข 1 และคอลัมน์อื่น ๆ เรียงตามลำดับ

0	4	0	0	9	0	6
4	0	0	7	0	0	0
0	0	0	0	0	1	3
0	7	0	0	2	0	0
9	0	0	2	0	0	0
0	0	1	0	0	0	4
6	0	3	0	0	4	0

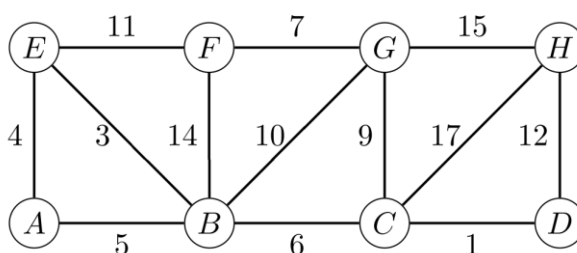
- a) (5 คะแนน) จงรัน Prim Algorithm โดยเริ่มต้นจากปม 1 และระบุลำดับของปมที่ถูกพิจารณาให้เป็น MST

- b) (5 คะแนน) จงรัน Kruskal's Algorithm และระบุลำดับเส้นที่ถูกนำมาใส่ใน minimum spanning tree (ระบุด้วยคํ่าันดับของปม)

4. (5 คะแนน) จงหา Topological Sort ของกราฟดังต่อไปนี้ หากมีหลายคำตอบ นิสิตตอบมาเพียงหนึ่งคำตอบก็พอ



5. (10 คะแนน) จงรัน Dijkstra's Algorithm บนกราฟข้างล่างนี้โดยเริ่มจากปม A ให้ระบุลำดับของปมที่ถูกพิจารณาและระยะทางระหว่างการเดินทางคำนวณในแต่ละรอบการทำงานของ Dijkstra's Algorithm

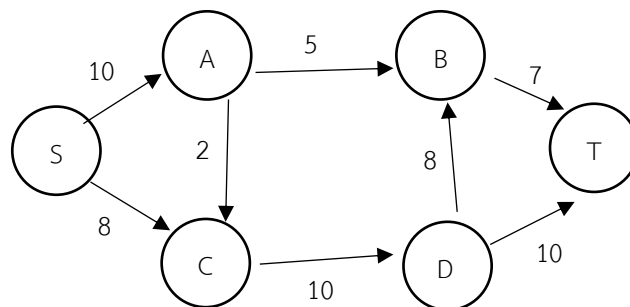


a) (5 คะแนน) จงระบุลำดับของปมที่ถูกพิจารณา

b) (5 คะแนน) จงระบุระยะทางสั้นสุดไปยังแต่ละปมในแต่ละรอบการทำงาน

รอบที่	dist[A]	dist[B]	dist[C]	dist[D]	dist[E]	dist[F]	dist[G]	dist[H]
0	0	∞	∞	∞	∞	∞	∞	∞

6. (10 คะแนน) จงหา Maximum Flow ของกราฟดังต่อไปนี้ด้วย Ford-Fulkerson Algorithm พร้อมแสดง Residue Graph หลังจบการทำงาน



(a) (3 คะแนน) ขนาดของ Maximum Flow คือ _____

(b) (7 คะแนน) Residue Graph หลังจบการทำงานคือ

7. (5 คะแนน) ให้ Matrix $W[i][j]$ เก็บความยาวของเส้น i ไปยัง j โดย $1 \leq i, j \leq n$ ให้นิยามเขียน Pseudo Code ของ Floyd-Warshall algorithm เพื่อหาระยะทางที่สั้นที่สุดจาก ทุกๆ คู่ปม ใส่ใน Matrix $D[i][j]$

8. (10 คะแนน) โดมิโนเป็นแท่งหินสี่เหลี่ยม เราสามารถเอาโดมิโนมาวางใกล้ ๆ กัน แล้วพลิกให้โดมิโนตัวหนึ่งล้มลง เมื่อโดมิโนล้มลง โดมิโนนั้นก็จะไปชนโดมิโนอื่นให้ล้มด้วย เกมโดมิโนคือการวางแท่งโดมิโนต่อกันไปเรื่อย ๆ แล้วพลิกให้โดมิโนล้มให้หมด ถ้าเราวางโดมิโนได้ดี เราจะพลิกโดมิโนแค่ชิ้นเดียวแล้วโดมิโนดังกล่าวจะล้มไปโดนชิ้นอื่น ๆ ทับกันไปเรื่อย ๆ จนโดมิโนล้มทั้งหมด แต่ถ้าเราวางไม่ดี เราอาจจะต้องช่วยพลิกโดมิโนให้ตัวอื่นล้มลงไปด้วย กำหนดให้มีโดมิโน n อัน แต่ละอันกำกับด้วยหมายเลข 1 ถึง n และให้ $\text{vector} \langle \text{pair} \langle \text{int}, \text{int} \rangle \rangle v$ เป็นรายการของการเรียงโดมิโนของเราโดยที่ $v[i].\text{first}$ จะเก็บหมายเลขโดมิโนที่ เมื่อล้มลงแล้ว $v[i].\text{second}$ จะล้มตามไปด้วย

จากข้อมูลการวางโดมิโนที่กำหนดให้ จงออกแบบอัลกอริทึมที่คำนวณว่า เราจะต้องช่วยใช้มือพลิกโดมิโนให้ล้มเป็นจำนวนครั้งให้น้อยที่สุดกี่ครั้ง โดมิโนถึงจะล้มหมด ตัวอย่างเช่นให้ $v = \langle \{2,1\}, \{2,4\}, \{3,2\} \rangle$ ถ้าเราพลิกโดมิโน 3 ให้ล้ม 3 จะไปชน 2 ให้ล้ม และ 2 จะไปชน 1 และ 4 ให้ล้มตาม ดังนั้น สำหรับข้อมูลนี้ เราต้องพลิกเพียง 1 ครั้ง แต่ถ้า $v = \langle \{1,2\}, \{3,4\} \rangle$ เราต้องพลิก 1 และ 3 ให้ล้มด้วยมือ โดมิโนถึงจะล้มทั้งหมด ในกรณีนี้ต้องตอบ 2

การออกแบบอัลกอริทึมนั้น สามารถเรียกใช้อัลกอริทึมต่าง ๆ ที่ได้เรียนมาในได้ทั้งหมด โดยจะต้องระบุ input/output ของอัลกอริทึมที่เรียกใช้ให้ชัดเจน พร้อมทั้งระบุประสิทธิภาพเชิงเวลาด้วย

9. (10 points) Inheritance Distribution Problem (ปัญหาการแบ่งสมบัติให้ได้ใกล้เคียงกันที่สุด): One rich man has just passed away. He has n houses that must be given to his two children. Each house must be given to either child. We cannot divide any house. The house are labeled by the number 1 to n and the house numbered k has the value of $v[k]$. The Inheritance Distribution Problem is to distribute these n houses to the children such that the summation of the value of the houses given to one child is as close as possible to the value of those given to the other child. For example, assume that we have 3 houses, valued 2, 6 and 3. The best way to

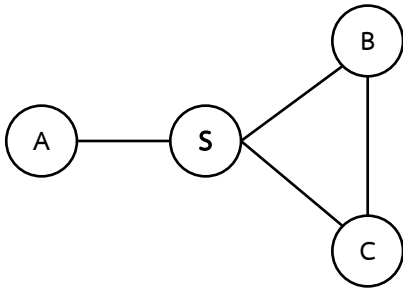
distribute these house is to have the first child get the first and the last house with the total value of 5 and let the other child get the second house valued 6. The different is only 1 which is the best that we could do.

Derive a state space search algorithm to solve this problem by answering the following question

- 9.1 (2 points) Describe the search space of this problem. What is the member of the search space? What is the size of the search space with respect to n ?
- 9.2 (5 points) The Branch-and-bound technique can be used to solve this problem. Design a heuristic function for branch-and-bound. Also, analyze the time complexity of the heuristic.
- 9.3 (3 points) Regardless of the earlier problem, let us assume that we have $h(x)$ which is a heuristic for this problem that predicts the “best possible value” of the state x . Write a pseudo-code of the state space search algorithm that solve the problem using $h(x)$.

10. (10 คะแนน) ปัญหา Longest Path เป็นดังต่อไปนี้ กำหนดให้มีกราฟ G ซึ่งเป็น undirected weighted simple graph และมี s เป็นปมปมหนึ่งในกราฟนี้ เราต้องการหา path ที่เริ่มต้นที่ปม s ที่ไม่ผ่านปมใดมากกว่า 1 ครั้ง ที่ผลรวมของน้ำหนักใน path ดังกล่าวนั้นมีค่ามากที่สุด จงแก้ปัญหานี้ด้วยวิธีการค้นในปริภูมิสถานะ โดยให้ตอบคำถามต่อไปนี้

10.1 (2 คะแนน) จงอธิบายปริภูมิสถานะของปัญหานี้ และระบุขนาดของปริภูมิสถานะตามขนาดของกราฟ และให้แจกแจงสมาชิกทั้งหมดของปริภูมิสถานะ สำหรับกราฟดังรูปต่อไปนี้



10.2 (3 คะแนน) จงระบุวิธีการ backtracking ที่ดีของปัญหานี้

10.3 (5 คะแนน) จงระบุ heuristic ที่ดีสำหรับปัญหานี้ พร้อมทั้งยกตัวอย่าง input โดยละเอียดที่ heuristic ดังกล่าวสามารถลดเวลาในการทำงานได้ โดยให้แสดงให้ดูถึงสถานะที่ heuristic ดังกล่าวจะใช้ประโยชน์ได้

11. (10 คะแนน) จงพิจารณาปัญหาการตัดสินใจ ซึ่งเกี่ยวข้องกับ การหาตัวประกอบ และ การทดสอบว่าตัวเลขเป็นจำนวนเฉพาะหรือไม่ ดังต่อไปนี้

11.1 จงเขียนรหัสเทียมเพื่อทวนสอบคำตอบว่าจำนวนเต็ม N มีตัวประกอบ d อยู่จริงหรือไม่ โดยให้เขียนฟังก์ชัน

`IsValid_IntegerFactor(N, M)` ซึ่งจะคืนค่า “True” ก็ต่อเมื่อ N มีตัวประกอบเป็นจำนวนเต็มที่มีค่า มากกว่า 1 แต่ไม่เกิน M และให้คืน “False” ในกรณีอื่น ๆ

11.2 จงออกแบบอัลกอริทึมแบบการค้นในปริภูมิสถานะ เพื่อหาเซตของจำนวนตัวเต็มที่ คูณกันได้ N ทุกรูปแบบที่เป็นไปได้ โดยให้เขียนฟังก์ชัน `Factor(N)` ซึ่งจะคืนเซตดังกล่าวทั้งหมด ตัวอย่างเช่น ถ้าเรียก `Factor(12)` ต้องได้คำตอบเป็น $\{12\}$, $\{2,6\}$, $\{2,2,3\}$ และ $\{3,4\}$

11.3 จงเขียนรหัสเทียม เพื่อลดรูป (reduce) ปัญหาการตัดสินใจ N เป็นจำนวนเฉพาะหรือไม่ โดยให้แปลงปัญหาดังกล่าวเป็นปัญหา `IsValid_IntegerFactor(N, M)` กล่าวคือ ให้เขียนฟังก์ชัน `IsNotPrime(N)` ซึ่งจะคืนค่า “True” ก็ต่อเมื่อ N “ไม่เป็น” จำนวนเฉพาะเท่านั้น และให้คืนค่า “False” ในกรณีอื่น ๆ โดยในฟังก์ชันนี้ สามารถเรียกใช้ฟังก์ชัน `IsValid_IntegerFactor(N, M)` ซึ่งได้นิยามไว้ในข้อ 11.1 ได้

12. (10 คะแนน) ปัญหา Feedback Edge Set เป็นดังต่อไปนี้ กำหนดให้มี undirected weighted simple graph G โดยที่ปมแต่ละปมกำกับด้วยหมายเลข 1 ถึง n และให้ $w[i][j]$ คือน้ำหนักของเส้นเชื่อมที่เชื่อมปม i และปม j เส้นเชื่อมในกราฟนี้มีน้ำหนักเป็นจำนวนเต็มบวกทั้งหมด ถ้าหากไม่มีเส้นเชื่อมที่เชื่อมปม i และ j แล้ว $w[i][j]$ จะเป็น 0

เราต้องการเลือกเส้นเชื่อมของกราฟนี้มาจำนวนหนึ่ง โดยเส้นเชื่อมที่เลือกนั้นต้องมีเงื่อนไขดังต่อไปนี้ 1) เมื่อลบเส้นเชื่อมที่เลือกทั้งหมดออกจากกราฟแล้ว กราฟที่เหลือจะต้องไม่มี cycle 2) ผลรวมของน้ำหนักของเส้นเชื่อมที่เลือกมีค่าน้อยที่สุด

ขอเสนออัลกอริทึมหนึ่งสำหรับปัญหานี้ดังต่อไปนี้ ซึ่งดัดแปลงมาจากอัลกอริทึมของ Kruskal

- 1) กำหนดให้ S เป็นเซตว่าง
- 2) ให้เรียงเส้นเชื่อมทั้งหมดจากน้ำหนักมากไปน้อย (ตรงกันข้ามกับ Kruskal's ที่เรียงจากน้อยไปมาก)
- 3) ค่อย ๆ พิจารณาเลือกเส้นเชื่อมทีละเส้นมาไว้ในเซต S ตามลำดับที่เรียงไว้ในข้อ 2) ตั้งแต่เส้นแรกถึงเส้นสุดท้าย
 - a. ถ้าเส้นเชื่อมที่เลือกเข้ามานั้น เมื่อนำมารวมกับ S แล้ว ไม่ก่อให้เกิด Cycle ก็เอาเส้นเชื่อมดังกล่าวเพิ่มเข้าไปใน S
 - b. แต่ถ้าเส้นเชื่อมดังกล่าวก่อให้เกิด Cycle ก็ไม่ต้องเอาเส้นเชื่อมดังกล่าวเพิ่มเข้าไปใน S ให้ข้ามเส้นนั้นไป
- 4) คำตอบของปัญหานี้คือเส้นเชื่อมทั้งหมด “ที่ไม่อยู่” ใน S (ซึ่งก็คือเส้นเชื่อมที่เราข้ามมาทั้งหมด)

จงอธิบายว่า วิธีการดังกล่าว สามารถแก้ปัญหา Feedback Edge Set ได้หรือไม่ ถ้าแก้ไม่ได้ ให้ยกตัวอย่างกราฟที่ทำงานแล้วได้ผลลัพธ์ไม่ถูกต้องมา แต่ถ้าอัลกอริทึมนี้สามารถแก้ปัญหานี้ได้ ให้อธิบายให้เห็นจริง ชัดแจ้งว่าทำไมถึงเป็นเช่นนั้น