FACULTY OF ENGINEERING

CHULALONGKORN UNIVERSITY

2110327 Algorithm Design

YEAR III, First Semester, Final Examination, September 29, 2011, Time 8:30 – 11:30

ชื่อ-นามสกุล_____เลขประจำตัว | | | | | | | | | 2 | 1 | CR58_____

หมายเหตุ

1. ข้อสอบมีทั้งหมด 3 ตอน รวม 23 ข้อในกระดาษคำถามคำตอบจำนวน 3 แผ่น 6 หน้า คะแนนเต็ม 72 คะแนน
2. ไม่อนุญาตให้นำตำราและเครื่องคำนวณต่างๆ ใดๆ เข้าห้องสอบ
3. ควรเขียนตอบด้วยลายมือที่อ่านง่ายและชัดเจน สามารถใช้ดินสอได้
4. ห้ามการหยิบยืมสิ่งใดๆ ทั้งสิ้น จากผู้สอบอื่นๆ เว้นแต่ผู้คุมสอบจะหยิบยืมให้
5. ห้ามนำส่วนใดส่วนหนึ่งของข้อสอบออกจากห้องสอบ ข้อสอบเป็นทรัพย์สินของราชการซึ่งผู้ลักพาอาจมีโทษทางคดีอาญา
6. ผู้ที่ประสงค์จะออกจากห้องสอบก่อนหมดเวลาสอบ แต่ต้องไม่น้อยกว่า 45 นาที
7. เมื่อหมดเวลาสอบ ผู้เข้าสอบต้องหยุดการเขียนใดๆ ทั้งสิ้น
8. ผู้ที่ปฏิบัติเข้าข่ายทุจริตในการสอบ ตามประกาศคณะวิศวกรรมศาสตร์ มีโทษ คือ ได้รับ สัญลักษณ์ F ในรายวิชาที่ทุจริต และพักการศึกษาอย่างน้อย 1 ภาคการศึกษา

รับทราบ

ลงชื่อนิสิต (.....)

หมายเหตุ (เพิ่มเติม)

1. ข้อใดที่ให้ออกแบบอัลกอริทึมนั้น นิสิตสามารถตอบโดยเขียนบรรยายแนวคิดที่ implement ได้ในทางปฏิบัติ หรือจะเขียนเป็นรหัสเทียมประกอบแนวคิดที่นำเสนอด้วยก็ได้
2. ต้องแสดงวิธีทำทุกข้อ การเขียนคำตอบเพียงอย่างเดียวจะไม่มีคะแนนให้ (ยกเว้นว่าจะเขียนในคำสั่ง)
3. ให้นิสิตเขียนรหัสประจำตัวและเลขที่ใน CR58 **<u>ในทุกหน้าของกระดาษคำถามด้วย</u>**
4. ตอนที่ 1 และ 2 ให้ทำในกระดาษคำถาม ตอนที่ 3 ให้ทำในสมุดคำตอบ สำหรับตอนที่ 3 ให้เขียนตอบข้อที่ k ไว้ที่หน้าที่ 2k -1 และ 2k ในสมุดคำตอบ (k = 1, 2, 3, 4, 5)
5. นิสิตสามารถใช้ขั้นตอนวิธีหรือโครงสร้างข้อมูลใด ๆ ที่อยู่ในเนื้อหาได้โดยตรง ไม่จำเป็นต้องเขียนขึ้นมาใหม่ แต่จำเป็นที่จะต้องระบุให้ชัดว่าใช้งานอย่างไร
6. กรณีที่มีข้อสงสัยใด ๆ หรือไม่แน่ใจในข้อกำหนด ให้เขียนระบุข้อสมมติฐานที่นิสิตใช้ลงไปในคำตอบด้วย

ตอนที่ 1 (รวม 12 คะแนน)

จงพิจารณาข้อความในแต่ละข้อต่อไปนี้ว่า เป็นข้อความที่จริงหรือเท็จ (ไม่ต้องอธิบาย) ถ้าคำตอบที่เขียนเป็นคำตอบที่ถูกต้องได้ 1 คะแนน แต่ถ้าคำตอบที่เขียนเป็นคำตอบที่ผิดเสีย 1 คะแนน (เขียนคำว่า **จริง** หรือ **เท็จ** ที่หน้าหมายเลขข้อ)
(ทำให้กระดาษคำถามแผ่นนี้)

1. เราสามารถหา longest simple path ของ directed acyclic graph ได้ในเวลา O(v + e) v คือจำนวน vertices และ e คือจำนวน edges ของกราฟ
2. directed acyclic graph ที่มี longest path ยาว v-1 ย่อมมี topological sort เพียงแบบเดียว (v คือจำนวน vertices)
3. ข้างล่างนี้ก็คืออัลกอริทึมของ Bellman-Ford เพื่อหาระยะทางของเส้นทางสั้นสุดจาก s ถึง t ในกราฟ G นั่นเอง

```
ShortestPath( w[1..n][1..n], s, t ) {
  d = new array of size n
  for (i=1 to n) d[i] = ∞;
  d[s] = 0
  for (k=1 to n-1)
    for (i=1 to n)
      for (j=1 to n)
        if ( d[i]+w[i][j] < d[j] ) d[j] = d[i]+w[i][j]
  return d[t];
}
```

4. G เป็น directed graph มี s เป็น vertex เดียวเท่านั้นที่มีเส้นเชื่อมพุ่งออกที่ความยาวเป็นลบ ถ้าใช้ Dijkstra's algorithm หา shortest paths จาก s ถึงปมอื่น ๆ ใน G จะได้คำตอบถูกต้องแน่นอน
5. เราสามารถหาจำนวน components ของกราฟได้ในเวลา O(v) โดยที่ v คือจำนวน vertices ของกราฟ
6. ให้ G คือ directed graph ถ้าการทำ depth-first search ใน G ใช้เวลา T_D และการทำ breadth-first search ใน G ใช้เวลา T_B จะได้ $T_D > T_B$ เสมอ
7. การค้นคำตอบของปัญหา sum of subset ที่มีเซตขนาด 1,000 ตัว ด้วย depth-first search นั้นนอกจากจะช้ามาก ๆ อาจเกิดปัญหาหน่วยความจำไม่พอก็เป็นได้ เนื่องจาก state space ใหญ่มาก ๆ
8. backtracking เป็นกลไกเพื่อเลือก state ที่เหมาะสมในการเข้าสู่คำตอบได้รวดเร็วขึ้น
9. bin packing คือปัญหาในการนำสิ่งของขนาดต่าง ๆ บรรจุลงในถัง โดยให้ใช้จำนวนถังน้อยสุด หากต้องการแก้ปัญหานี้ด้วยกลวิธีการค้นแบบ branch-and-bound ต้องรู้วิธีการคำนวณ lower bound ของจำนวนถังที่ใช้บรรจุของ ณ state ต่าง ๆ ใน state space
10. decision problem ที่เป็นปัญหา P ย่อมเป็นปัญหา NP อย่างแน่นอน
11. ปัญหา NP-complete เป็นปัญหาที่ verify คำตอบจริงได้ด้วยอัลกอริทึมที่ทำงานในเวลาแบบ polynomial
12. การพิสูจน์ว่าแท้จริงแล้วเซตของปัญหา P ไม่เท่ากับเซตของปัญหา NP เป็นเรื่องที่คนในวงการคาดกันว่าน่าจะเป็นเช่นนั้น

ตอนที่ 2 (รวม 10 คะแนน)

จงออกแบบอัลกอริทึม โดยบรรยายแนวคิดสั้น ๆ เขียนให้อ่านง่ายได้ใจความ ในที่ว่างที่เตรียมให้ ควรอ้างอิงการใช้อัลกอริทึมที่รู้จักให้เป็นประโยชน์ (ให้ v แทนจำนวน vertices และ e แทนจำนวนเส้นเชื่อมในกราฟที่กำหนดให้ในโจทย์) (ไม่ต้องวิเคราะห์ประสิทธิภาพ)
เลือกทำได้ 2 ข้อจาก 6 ข้อข้างล่างนี้ ถ้าทำเกิน 2 ข้อ จะตรวจเฉพาะข้อที่คำตอบมีหมายเลขข้อน้อย 2 ข้อแรกเท่านั้น

1. G เป็น undirected graph ที่เส้นเชื่อมยาว 1 เท่ากันหมดทุกเส้น จงออกแบบอัลกอริทึมที่มีประสิทธิภาพสุด ๆ เพื่อหาจำนวน shortest paths จาก a ไป b ใน G ว่ามีกี่เส้นทางที่แตกต่างกันทั้งหมด

2. G เป็น directed graph ที่ความยาวเส้นเชื่อมเป็นจำนวนบวกทั้งหมด และมี cycle อยู่ภายใน จงออกแบบอัลกอริทึมที่ใช้เวลา $O(v^3)$ เพื่อหาความยาวของ cycle ที่สั้นสุดใน G

3. G เป็น connected undirected graph จงออกแบบอัลกอริทึมที่ใช้เวลาที่มีประสิทธิภาพสุด ๆ เพื่อหาว่ามี edge ใน G ไหม ที่เมื่อลบออกแล้ว G ยังคง connected

4. ให้ G คือ undirected graph และ(a,b) เป็นเส้นเชื่อมหนึ่งของ G จงออกแบบอัลกอริทึมที่มีประสิทธิภาพเชิงเวลาสุด ๆ เพื่อหาว่า G มี cycle ใดหรือไม่ ที่มีเส้นเชื่อม (a,b) เป็นส่วนหนึ่งของ cycle

5. คณะวิศวฯ ตัดสินใจเปิดหลักสูตร วิศวสำอาง ขึ้นมาใหม่ มีแต่วิชาบังคับ ผู้ออกแบบหลักสูตรเขียนเงื่อนไข pre-requisite ของวิชาต่าง ๆ เป็นกราฟ มี vertex แทนวิชา และเส้นเชื่อมจาก a ไป b แทนเงื่อนไขที่ต้องเรียน a ให้ผ่านก่อนเรียน b สมมติว่านิสิตสามารถเรียนวิชาได้ไม่จำกัดในหนึ่งภาคการศึกษา จงออกแบบอัลกอริทึมเพื่อหาว่า นิสิตสามารถสำเร็จการศึกษาในหลักสูตรนี้ได้เร็วสุดในเวลากี่ภาคการศึกษา

6. ให้ G เป็น undirected graph จงออกแบบอัลกอริทึมเพื่อหาว่า G มี cycle ที่ประกอบด้วยเส้นเชื่อม 4 เส้นหรือไม่ (บรรยายแนวคิดสั้น ๆ ข้อนี้ขออัลกอริทึมที่ใช้เวลา $O(v^3)$ ก็ได้)

ตอนที่ 3 (ข้อละ 10 คะแนน)

ข้อต่อไปนี้เป็นการออกแบบอัลกอริทึม ให้นิสิตวิเคราะห์ประสิทธิภาพเชิงเวลาด้วยทุกข้อ และคะแนนของแต่ละข้อนั้นจะขึ้นอยู่กับประสิทธิภาพของอัลกอริทึมด้วย

1. กำหนดให้ G เป็น undirected weighted connected graph โดยที่แต่ละ edge นั้นมีความยาวเป็น 1 เรานิยาม Diameter ของ G คือค่าที่มากที่สุดของระยะทางสั้นสุดของทุก ๆ คู่ปมในกราฟดังกล่าว หรือกล่าวอีกอย่างได้ว่า ถ้าให้ d(a,b) เป็นระยะทางสั้นสุดจาก vertex a ไปยัง vertex b ค่า Diameter คือ Max(d(a,b)) เมื่อ a,b เป็น vertex ใด ๆ ในกราฟ G จงออกแบบอัลกอริทึมสำหรับการหาค่า Diameter เมื่อมี G = (V,E) เป็นข้อมูลนำเข้า (ข้อสังเกต: d ไม่ใช่ข้อมูลนำเข้า)
2. กำหนดให้ “เกมบันไดงู CP” เป็นดังต่อไปนี้ มีช่องเรียงต่อกันเป็นแถวอยู่ 100 ช่อง แต่ละช่องถูก indexed ด้วยหมายเลข 0 ถึง 99 ผู้เล่นจะเริ่มที่ช่อง 0 และเป้าหมายคือการไปถึงช่อง 99 ด้วยค่าใช้จ่ายน้อยสุด การเดินจากช่อง a ไปช่อง b จะ “เสียเงิน” เท่ากับ b – a บาท อย่างไรก็ตาม จะมีช่องพิเศษที่เรียกว่า “บันไดงู” อยู่ n ช่อง (indexed ด้วยหมายเลข 0 ถึง n-1) โดยที่บันไดงูแต่ละอันจะอยู่ที่ช่อง s[i] บันไดงูที่ s[i] นั้นจะมี “งู” เชื่อมย้อนไปยังช่อง t[i] โดยที่ s[i] > t[i] เสมอ ถ้าผู้เล่นอยู่ที่ช่อง s[i] ผู้เล่นสามารถเลือกที่จะ “ไต่งู” ย้อนไปยังช่อง t[i] ได้โดยจะ “ได้รับ” เงินจากงูเป็นจำนวน p[i] บาท ถ้ากำหนดให้ในการเล่น “เกมบันไดงู CP” นี้ ผู้เล่นสามารถเลือกไต่งูได้ไม่เกิน k ครั้ง และบันไดงูแต่ละอันใช้ได้เพียงครั้งเดียว จงออกแบบอัลกอริทึมสำหรับหาว่าผู้เล่นควรจะใช้บันไดงูที่ช่องใดบ้าง เมื่อกำหนดให้ข้อมูลนำเข้าเป็น n, s[], t[], p[], และ k ซึ่งมีค่าเป็นจำนวนเต็มไม่ลบทั้งหมด

1. คุณต้องการสร้างหุ่นยนต์ตัวหนึ่งซึ่งประกอบด้วยชิ้นส่วนย่อยจำนวน n อย่าง (แต่ละอย่าง indexed ด้วยหมายเลข 0 ถึง n-1) และมีร้านขายอุปกรณ์หุ่นยนต์อยู่ m ร้าน (แต่ละร้าน indexed ด้วยหมายเลข 0 ถึง m-1) ให้ S[a][b] เป็นบูลีนเมทริกซ์ขนาด n x m โดยที่ S[a][b] จะมีค่าเป็นจริงก็ต่อเมื่อร้านที่ a ขายอุปกรณ์อย่างที่ b เราต้องการทราบจำนวนร้านน้อยที่สุดที่คุณต้องไปซื้อของ (ของทุกอย่างมีขายอยู่อย่างน้อย 1 ร้าน) ให้คุณ

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

   ก. ออกแบบอัลกอริทึมค้นหา (search algorithm) เพื่อแก้ปัญหานี้
   ข. จงหาเงื่อนไขในการทำ backtracking หรือ bounding heuristic (bounding function) สำหรับปัญหานี้ และยกตัวอย่างกรณีที่ backtracking หรือ bounding heuristic ดังกล่าวสามารถใช้ประโยชน์ได้
   ค. เขียน state space tree ของอัลกอริทึมในข้อ ก. และให้ระบุปมที่เกิดการ backtracking หรือ bounding ด้วย เมื่อกำหนดให้ n = 6, m = 5 และ เมทริกซ์ S เป็นดังตารางด้านขวามือ (เลข 1 ในแถว a คอลัมน์ b หมายถึงร้าน a มีขายของอย่างที่ b)
2. วิชาการออกแบบอัลกอริทึมสำหรับปี 2555 นั้นมีการบ้านอยู่จำนวน n การบ้าน (indexed ด้วยหมายเลข 0 ถึง n-1) สมมติว่าคุณทราบว่าการบ้านชิ้นที่ i นั้นมีกำหนดส่ง ณ วันที่ d[i] และจะได้คะแนน s[i] คะแนนเมื่อทำการบ้านเสร็จ การส่งการบ้านดังกล่าวหลังวันที่ d[i] จะเสียคะแนนวันละ p[i] คะแนน คุณทราบว่าสำหรับการบ้านชิ้นที่ i นั้น คุณจะใช้เวลาทำการบ้าน t[i] พอดี (คุณไม่สามารถทำการบ้านมากกว่า 1 ชิ้นได้ในช่วงเวลาเดียวกัน) อย่างไรก็ตาม วิชานี้มีกฎพิเศษอยู่ดังนี้
   - คุณสามารถที่จะเลือก “ทำ” หรือ “ไม่ทำ” การบ้านแต่ละข้อได้
   - การเลือก “ไม่ทำ” การบ้าน i หมายความว่าคุณจะไม่ได้คะแนน และ จะไม่เสียคะแนนจากการบ้านดังกล่าว

- การเลือก “ทำ” นั้นคุณจะต้องทำการบ้านนั้นจนเสร็จและส่งการบ้านนั้นก่อนที่จะไปเลือก “ทำ” หรือ “ไม่ทำ” ข้ออื่น
- ถ้าคุณเลือก “ทำ” การบ้านชิ้นที่ i แล้ว การบ้านชิ้นถัดไปที่จะเลือก “ทำ” ได้คือการบ้านชิ้นที่ i+1 ถึงชิ้นที่ n-1 เท่านั้น (เช่น ถ้าคุณเลือกทำการบ้านชิ้นที่ 5 แล้ว จะย้อนกลับไปทำชิ้นที่ 3 ไม่ได้ เป็นต้น)

จงออกแบบอัลกอริทึมสำหรับหาว่า การเลือกทำการบ้านแบบที่ได้คะแนนมากสุดนั้นจะได้คะแนนเท่าไร เมื่อกำหนดให้ n, d[], s[], p[], และ t[] เป็นข้อมูลนำเข้า ซึ่งมีค่าเป็นจำนวนเต็มไม่ลบทั้งหมด (สำหรับข้อนี้ คำตอบที่ได้คะแนนเต็มจะต้องใช้เวลาเป็น $O(n^2)$ หรือดีกว่า)

3. กำหนดให้มีเมืองอยู่ n เมือง (indexed ด้วยหมายเลข 0 ถึง n-1) และกำหนดให้เมืองที่ i ตั้งอยู่บนแผนที่ ณ พิกัด (x[i],y[i]) กำหนดให้เวลาที่ใช้ในการเดินทางระหว่างเมือง a และเมือง b เท่ากับระยะห่างระหว่างเมือง a และเมือง b ซึ่งก็คือรากที่สองของ $(x[a] - x[b])^2 + (y[a] - y[b])^2$ กำหนดให้เราเดินทางด้วยความเร็ว 1 หน่วยต่อ 1 วินาที ปัจจุบันเราอยู่ที่เมือง 0 และเราต้องการไปเมือง n-1 ซึ่งแน่นอนว่าเส้นทางที่เร็วที่สุดคือการเดินทางเป็นเส้นตรง อย่างไรก็ตาม ที่เมืองแต่ละเมืองนั้นมี time machine อยู่ ซึ่ง time machine ที่เมือง i นั้นจะพาเราย้อนเวลากลับไปในอดีตเป็นเวลา t[i] วินาทีก่อนหน้าโดยไม่เปลี่ยนสถานที่ และ time machine ของแต่ละเมืองนั้นสามารถใช้ได้เพียงครั้งเดียวต่อ 1 เมือง (ห้ามใช้ time machine ในเมืองเดียวกันเกิน 1 ครั้ง) จงออกแบบอัลกอริทึมเพื่อหาเวลารวมน้อยสุดที่เราสามารถเดินทางไปยังเมือง n – 1 ได้ เมื่อกำหนดให้ข้อมูลนำเข้าเป็น n, x[], y[] และ t[] ซึ่งมีค่าเป็นจำนวนเต็มไม่ลบทั้งหมด (สำหรับข้อนี้ คำตอบที่ได้คะแนนเต็มจะต้องใช้เวลาเป็น $O(n^4 2^{2n})$ หรือดีกว่า)