ชื่อ _____ รหัส _____ section ___

สอบกลางภาค: ภาคการศึกษาที่ 2 **ปีการศึกษา**: 2557

วันสอบ: 17 มีนาคม 2558 **เวลาสอบ**: 13.30 – 16.30 น.

ห้องสอบ: A401

ผู้สอน: อ.เสกสรรค์ สุวรรณมณี อ.อัมรินทร์ ดีมะการ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

รหัสและชื่อวิชา: 242-310 Introduction to Algorithm and Complexity

แนะนำขั้นตอนวิธีและความซับซ้อน

ทุจริตในการสอบมีโทษขั้นต่ำคือ ปรับตกในรายวิชาที่ทุจริตและพักการเรียน 1 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 17 **หน้า** (*รวมหน้าปก*) แบ่งออกเป็น 5 **ตอน** คะแนนรวม 140 คะแนน (คิดเป็นคะแนนเก็บ 35%)
- ☐ **เขียนคำตอบในข้อสอบ คำตอบส่วนใดอ่านไม่ออก จะถือว่าคำตอบนั้นผิด**
- ☐ อ่านคำสั่งในแต่ละข้อให้เข้าใจก่อนลงมือทำ
- ☐ เวลาที่ใช้เวลาทำตอนให้เหมาะสม ตามคำแนะนำ
- ☐ หากข้อใดเขียนคำตอบไม่พอ ให้เขียนเพิ่มที่ด้านหลังของหน้านั้นเท่านั้น

ตอน	**1** (20) 5%	**2** (32) 8%	**3** (32) 8%	**4** (24) 6%	**5** (32) 8%	**รวม** (140) 35%
คะแนน						

นักศึกษารับทราบ ลงชื่อ .....

ตอนที่ 1 (20 คะแนน, 5%, 20 นาที)

Introduction to Algorithm พื้นฐานอัลกอริทึม

1. จงอธิบายความหมายและคุณสมบัติของขั้นตอนวิธี หรือ อัลกอริทึม (algorithm) (5 คะแนน)

2. จงบรรยายปัญหาเชิงคำนวณต่อไปนี้ ด้วยลักษณะของข้อมูลขาเข้า (input) ผลลัพธ์ที่ต้องการ (output) ตัวอย่างปัญหา (problem instance) และ output ของตัวอย่างปัญหา และตัวกำหนดขนาดของปัญหา (problem size) เลือกทำข้อใดข้อหนึ่งเพียงข้อเดียว (7 คะแนน)

2.1) Permutation codes แสดงรหัสตัวเลข k ($k \geq 1$) หลักที่เป็นไปได้ทั้งหมด โดยรหัสประกอบด้วยตัวเลขตั้งแต่ 0 ถึง n ($0 \leq n \leq 9$)

2.2) More than average จงนับจำนวนข้อมูลทั้งหมดที่มีค่ามากกว่าหรือเท่ากับค่าเฉลี่ย จากข้อมูล n ตัว ในอาเรย์ A[1..n]

3. จงออกแบบอัลกอริทึมสำหรับแก้ปัญหาในข้อ 2 (เลือกเพียงปัญหาเดียว) เขียนเป็นรหัสเทียม (pseudo code) และวิเคราะห์หาฟังก์ชันเวลาในการทำงาน (running time) หรือ *T(N)* ของอัลกอริทึมนั้น (8 คะแนน)

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

ตอนที่ 2 (32 คะแนน, 8%, 40 นาที)

Algorithm Analysis การวิเคราะห์อัลกอริทึม

1. จงเขียนนิยามและความหมายของสัญกรณ์เชิงเส้นกำกับ (Asymptotic notations) ต่อไปนี้ Big-O, Big Omega (Ω), Big Theta (Θ) พอสังเขป พร้อมทั้งแสดงกราฟเส้นประกอบ (6 คะแนน)

2. True or False ถูกหรือผิด ให้ใส่เครื่องหมายถูก √ หรือ ผิด X หน้าข้อต่อไปนี้ (6 คะแนน)

_____ a) If $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$ then $g(n) \prec f(n)$, (f(n) grows faster than g(n))

_____ b) log $(2n^3 + 1) \prec 3n^2$

_____ c) If f(n) = O(g(n)) and f(n) = Ω(g(n)) then f(n) = Θ(g(n))

_____ d) if f(n) = Θ (g(n)) then g(n) = Θ (h(n))

_____ e) if f(n) = Ω(g(n)) and g(n) = Ω(h(n)) then f(n) = Ω (h(n))

_____ f) $2^n \neq \Theta(2^{n+3})$

3. จงพิสูจน์ว่า $3n^2 + n\log n = O(n^2)$ (5 คะแนน)

4. จงหา Big-O ของ running time *t(n)* หรืออัลกอริทึมต่อไปนี้ (15 คะแนน)

4.1) $t(n) = 2n^5 + 500\ (n^3) - 8$

4.2) $t(n) = a^n + n!$, $a > 1$

4.3) $t(n) = 2t(n/2) + n^2$

4.4)

```
mySort(A[1..n])
{ for (i=1 to n-1)
  { m=A[i]; mj=i
    j=i+1
    while(j<=n)
    { if (A[j]<m )
       { mj=j; m=A[j] }
    }
    swap(A[i], A[mj] )
  }
}
```

4.5)

```
searchBST(A[1..n], key)
{ h=0; p = 2^h
  while ( p<=n )
  {
     if (key = A[p])
       return p
     h++
     if (key <A[p])
       p = 2p //left child node
     else
       p = 2p+1  //right child
  }

  return -1 //not found
}
```

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

ตอนที่ 3 (32 คะแนน, 8%, 40 นาที)

Data Structures โครงสร้างข้อมูล

1. กำหนดโครงสร้างข้อมูลแบบตารางแฮช (Hash Table) ขนาด 19 สำหรับเก็บข้อมูลคำศัพท์จำนวนหนึ่ง ซึ่งประกอบด้วยตัวอักษรพิมพ์เล็ก a-z เท่านั้น

1.1) Hash function

กำหนดให้แฮชฟังก์ชัน *h* คำนวณค่า hash value (ซึ่งคือค่า index ใน Hash table) จากคำศัพท์ (word) โดยให้ใช้ค่าตัวอักษร 3 **ตัวแรก**ของคำศัพท์นั้น คำนวณหาผลบวก แล้ว mod ด้วย ขนาดของ hash table คือ 19 ค่าของตัวอักษร คือค่าระหว่าง 0-25 คิดตามลำดับตัวอักษร a – z (เช่น a=0, b=1, c=2, ...)

$h(w) = (w_1+w_2+w_3) \bmod 19$

w คือ คำขนาด n ตัวอักษร ประกอบด้วยตัวอักษร $w_1, w_2, w_3, \ldots, w_n$

เช่น $h(apple) = (0+15+15) \bmod 19 = 30 \bmod 19 =$ **11**

ตารางค่าตัวอักษร

a=0	b=1	c=2	d=3	e=4	f=5	g=6	h=7	i=8	j=9	k=10	l=11	m=12
n=13	o=14	p=15	q=16	r=17	s=18	t=19	u=20	v=21	w=22	x=23	y=24	z=25

จงคำนวณหาค่า hash value ของคำต่อไปนี้ bat, cat, rat, attempt, banana (5 คะแนน)

$h(bat)$ = _____

$h(cat)$ = _____

$h(rat)$ = _____

$h(attempt)$ = _____

$h(banana)$ = _____

1.2) Hash Table

จงเติมคำศัพท์ bat, cat, rat, attempt, banana ลงใน Hash Table ให้ถูกต้อง (2 คะแนน)

Index	Item		Index	Item		Index	Item		Index	Item
0			5			10			15	
1			6			11	**apple**		16	
2			7			12			17	
3			8			13			18	
4			9			14				

1.3) Collision Resolution

จงเพิ่มคำต่อไปนี้ `human, dragon` ลงใน Hash Table จากข้อ 1.2 ให้ถูกต้อง อธิบายการเกิดการชนกัน (collision) และให้ใช้วิธี Opening Addressing แบบ linear probing ในการแก้ปัญหา (4 คะแนน)

h(human) = _____ *h(dragon)* = _____

Index	Item		Index	Item		Index	Item		Index	Item
0			5			10			15	
1			6			11	**apple**		16	
2			7			12			17	
3			8			13			18	
4			9			14				

1.4) จากตาราง Hash Table ข้อ 1.3) จงอธิบาย ขั้นตอนในการค้นหาคำต่อไปนี้ apple, ant, dragon, dog มีการค้นหาข้อมูลที่ตำแหน่งใดบ้าง และมีการเปรียบเทียบคำกี่ครั้ง จึงจะได้คำตอบว่าค้นเจอหรือไม่เจอ (2 คะแนน)

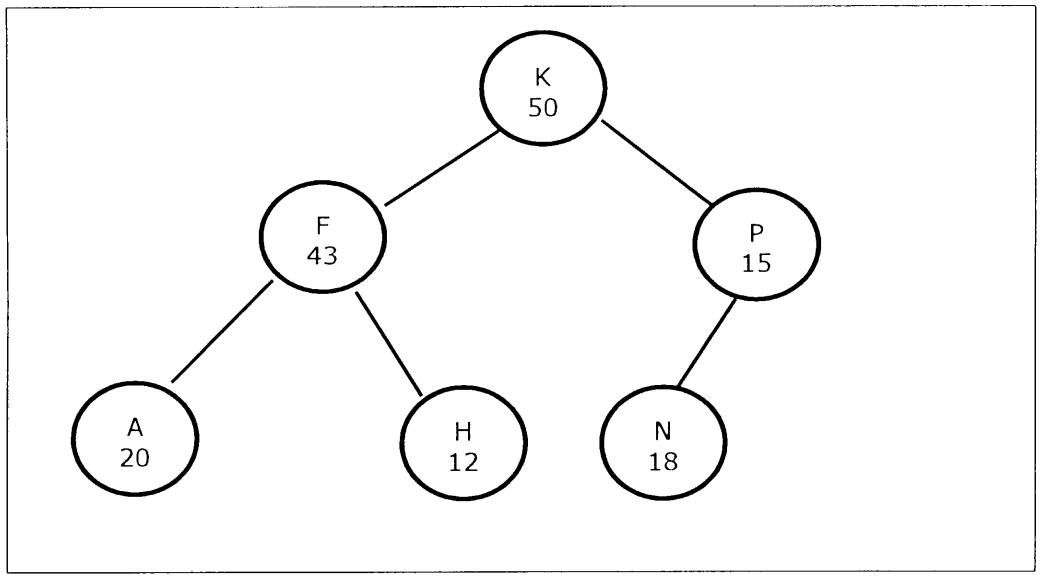
1.5) เวลาในการค้นหา ในกรณี Best case ใช้เวลาเพียง O(1) เกิดขึ้นเมื่อใด และกรณีทั่วๆไป การค้นหาใน Hash table ดีกว่าการค้นด้วยวิธี sequential search หรือไม่ อย่างไร (3 คะแนน)

2. โครงสร้างข้อมูล Treap (16 คะแนน)

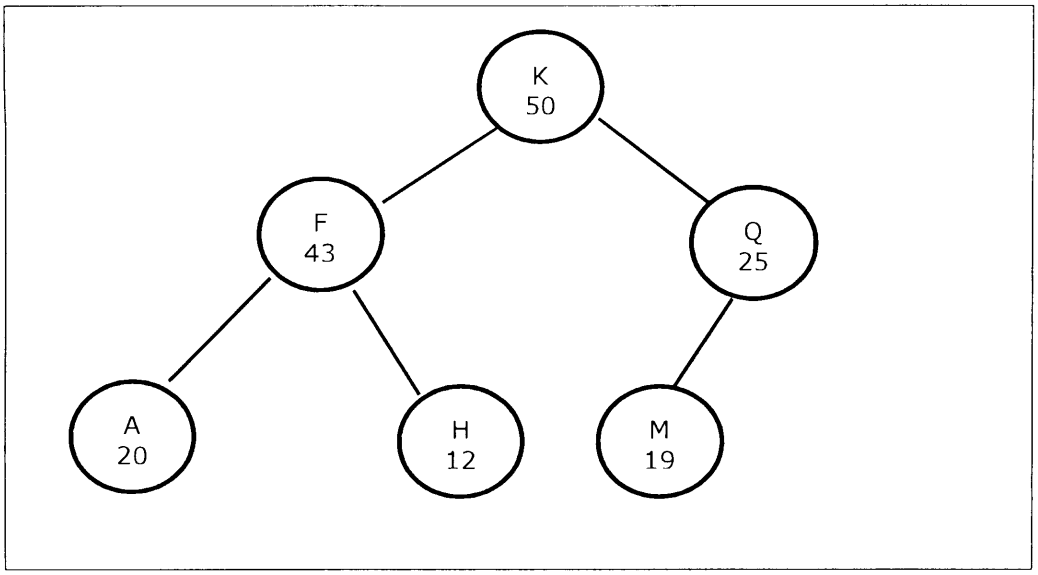
Treap คือโครงสร้างแบบ Binary Search Tree (BST) ที่เพิ่มคุณสมบัติของ Heap นั่นคือ แต่ละ node จะมีค่า Priority และ parent node จะมีค่า priority สูงกว่า children node เสมอ (Max Heap)

กำหนดให้ Node มีค่า key แทนด้วยตัวอักษร (เปรียบเทียบค่าตามลำดับตัวอักษร A-Z) และ ค่า priority กำหนดด้วยตัวเลขจำนวนเต็ม

2.1) Binary tree ดังรูป เป็น Treap หรือไม่ เพราะเหตุใด และมีวิธีการแก้ไข อย่างไร เพื่อทำให้เป็น Treap (วาดรูป Treap ใหม่ที่ได้) (4 คะแนน)



2.2) จาก Treap ดังรูปข้างล่าง ให้เพิ่มโหนดใหม่ ที่มีค่า key เป็น N และมี priority = 32 จากนั้น ทำการลบโหนด K (วาดรูปและอธิบายขั้นตอนการทำงาน) (6 คะแนน)



2.3) จงอธิบายอัลกอริทึม(เขียนเป็น pseudocode หรือภาษาธรรมชาติ) วิธีการการค้นหา (Search) การเพิ่มโหนด (Insert) และ การลบโหนดที่มี priority สูงสุด (RemoveMax) ใน Treap พร้อมทั้งสรุปความซับซ้อนเชิงเวลา (time complexity) ของแต่ละอัลกอริทึม ในรูป Big-O (6 คะแนน)

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

ตอนที่ 4 (24 คะแนน, 6 %, 40 นาที)

Searching and Sorting การค้นหาและการเรียงลำดับ

1. จากขั้นตอนวิธีในการเรียงลำดับข้อมูลที่กำหนดให้ต่อไปนี้ จงเลือกขั้นตอนวิธีที่เหมาะสมกับคำอธิบายในแต่ละข้อ

I. Bubble Sort II. Insertion Sort III. Merge Sort IV. Quick Sort

V. Selection Sort VI. Shell Sort VII. Heap Sort (6 คะแนน)

1.1 การเรียงลำดับข้อมูลที่ใช้หน่วยความจำ (space) เท่ากับ O(1)

1.2 เป็นการเรียงลำดับแบบการเรียงลำดับภายใน (Internal sort) (ตอบมาอย่างน้อย 3 คำตอบ)

1.3 เป็นขั้นตอนวิธีการเรียงลำดับแบบเสถียร (Stable sorting algorithms)

1.4 ขั้นตอนวิธีที่มีประสิทธิภาพในกรณี Worst case เป็น $O(N^{1.5})$

1.5 ขั้นตอนวิธีที่ใช้เวลาในการเปรียบเทียบเท่ากับ $O(N^2)$ และ เวลาในการสลับที่ เท่ากับ $O(N)$

1.6 ข้อมูลที่ต้องการเรียงลำดับมีขนาดใหญ่กว่าหน่วยความจำหลักที่จะเก็บได้หมด ข้อมูลส่วนใหญ่อยู่ในหน่วยความจำสำรอง เช่น hard disk

2. จงตอบคำถามหัวข้อขั้นตอนวิธีการเรียงลำดับต่อไปนี้ (9 คะแนน)

2.1 จงแสดงขั้นตอนของ *Selection Sort* ในการการเรียงข้อมูลต่อไปนี้ จากน้อยไปมาก และระบุเวลาที่ใช้ (time complexity, Big-O) ของขั้นตอนวิธีนี้ (4 คะแนน)

10, 21, 45, 8, 30, 53, 4, 50,2

2.2 เลือกใช้ขั้นตอนวิธีการเรียงลำดับข้อมูล(ที่ต่างจาก Selection Sort) ในแสดงขั้นตอนการเรียงลำดับข้อมูลนี้ ให้ระบุชื่อขั้นตอนวิธีและเวลาที่ใช้ (time complexity, Big-O) ของขั้นตอนวิธีที่เลือก (3 คะแนน)

62 , 83, 18, 53, 7, 17, 96, 85, 47, 69

2.3 จงอธิบายหลักการของขั้นตอนวิธีแบบ Merge Sort มาคร่าวๆ (2 คะแนน)

3. จงตอบคำถามต่อไปนี้ (9 คะแนน)

3.1 จงระบุเวลาที่ใช้(time complexity, Big-O) ในการค้นหาของ Sequential Search และ (1 คะแนน)

3.2 จงเขียน Pseudo Code ของขั้นตอนวิธีแบบ Binary Search (4 คะแนน)

3.3 จากข้อมูลในอาร์เรย์ที่กำหนดให้ จงตอบคำถามต่อไปนี้ (4 คะแนน)

Array Index	1	2	3	4	5	6	7	8	9	10
Key	-26	-5	- 3	1	6	10	20	60	40	89

1) จากข้อมูลที่กำหนดให้จงวาดต้นไม้เปรียบเทียบ(Comparison Tree) ของ Binary Search

2) จงหาจำนวนครั้งของการเปรียบเทียบเมื่อต้องการค้นหาเลข (key) 3 , 30 และ 60

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

ตอนที่ 5 (32 คะแนน, 8 %, 40 นาที)

Divide-and-Conquer การแบ่งแยกและเอาชนะ

1. การคำนวณเลขยกกำลัง (Power) จำนวนเต็ม (12 คะแนน)

Calculate a^n where $a > 0$ and $n \geq 1$

- Brute-force (naïve algorithm) : วิธีการตรงตัว (a คูณกัน n ตัว)

$$a^n = a * a * a * ... * a$$

1.1) จงเขียนอัลกอริทึม และหาเวลาการทำงาน (Big-O) ของการคำนวณเลขยกกำลังแบบวิธีการตรงตัว (4 คะแนน)

```
power(a, n)
```

- Divide-and-Conquer algorithm

$$a^n = \begin{cases} a^{n/2} * a^{n/2} & , n \text{ is even} \\ a * a^{(n-1)/2} * a^{(n-1)/2} & , n \text{ is odd} \end{cases}$$

1.2) จงเขียนอัลกอริทึมและหาเวลาการทำงาน (Big-O) ของการคำนวณด้วยวิธีการ divide-and-conquer โดยเขียนเป็นสมการเวียนเกิด (recurrence equation) และใช้ Master Method (8 คะแนน)

```
powerDC (a, n)
```

2. Matrix Multiplication การคูณเมตริกซ์ (12 คะแนน)

2.1) การคูณเมตริกซ์ ขนาด nxn สามารถใช้หลักการ Divide-and-Conquer ลดรูปการคูณให้เป็นการคูณของเมตริกซ์ขนาด (n/2)x(n/2) จำนวน 8 ครั้ง แล้วนำผลลัพธ์ทั้งหมดมารวมกัน โดยขั้นตอนการรวมใช้เวลา $O(n^2)$ จงเขียนสมการเวียนเกิด (recurrence equation) t(n) ของอัลกอริทึมนี้ แล้วหาผลลัพธ์เวลา ในรูป Big-O โดยใช้ Master Method (6 คะแนน)

2.2) Strassen's Algorithm เป็นการปรับปรุงการคูณเมตริกซ์ ข้หลักการ Divide-and-Conquer เช่นกัน แต่สามารถลดรูปการคูณให้เป็นการคูณของเมตริกซ์ขนาด (n/2)x(n/2) จำนวนเพียง 7 ครั้ง และนำผลลัพธ์ทั้งหมดมารวมกัน โดยขั้นตอนการรวมใช้เวลา $O(n^2)$ เช่นกัน

จงเขียนสมการเวียนเกิด (recurrence equation) t(n) ของ Strassen's Algorithm นี้ แล้วหาผลลัพธ์เวลา ในรูป Big-O โดยใช้ Master Method (6 คะแนน)

3. Selection Problem ปัญหาการเลือก (8 คะแนน)

จากปัญหาการเลือก (Selection Problem: select the k-th min) มีการใช้อัลกอริทึม QuickSelect ซึ่งใช้หลักการแบ่งส่วนข้อมูล(Partition) โดยอัลกอริทึม RandommizedPartition ใช้เวลาเป็น $\Theta(n)$ และมีสมการเวลา $t(n)$ ของอัลกอริทึมนี้ในลักษณะความสัมพันธ์เวียนเกิด (recurrence relation) ดังนี้

```
QuickSelect(A[left..right], m)
{ if (left = right) return A[left]
  j = RandomizedPartition( A[left..right) )
  k = j - left +1
  if (m <= k)
     return QuickSelect( A[left..j), m)
  else
     return QuickSelect( A[j+1..right), m-k)
}
```

กำหนดให้ $t(n)$ คือเวลาทำงานของ QuickSelect กับข้อมูล n ตัว

$$t(n) \leq \quad t\ (\ max\ (k,\ n\text{-}k)\ \ )\ +\ \Theta(n)$$

จากความสัมพันธ์ดังกล่าวจงเขียนสมการเวียนเกิด ในกรณีที่ดีที่สุด(best-case analysis) และ กรณีที่แย่ที่สุด (worst-case analysis) ของอัลกอริทึมนี้ และวิเคราะห์หาเวลาการทำงาน

(*Hint คำใบ้:* กรณีที่ดีที่สุดคือเมื่อสามารถแบ่งข้อมูลออกเป็นครึ่งๆเท่าๆกันได้ทุกครั้ง ส่วนเวลาที่แย่ที่สุดคือข้อมูลลดลงครั้งละหนึ่งเท่านั้น)

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