

FACULTY OF ENGINEERING  
CHULALONGKORN UNIVERSITY  
2110211 Introduction to Data Structures

YEAR II, Second Semester, Midterm Examination, March 3, 2012, Time 13:00 – 16:00

ชื่อ-นามสกุล \_\_\_\_\_ เลขประจำตัว 

								2	1
--	--	--	--	--	--	--	--	---	---

 CR58 \_\_\_\_\_

หมายเหตุ

1. ข้อสอบมีทั้งหมด 11 ข้อในกระดาษคำถามคำตอบจำนวน 7 แผ่น 7 หน้า คะแนนเต็ม 94 คะแนน
2. ไม่อนุญาตให้นำตำราและเครื่องคำนวณต่างๆ ใดๆ เข้าห้องสอบ
3. ควรเขียนตอบด้วยลายมือที่อ่านง่ายและชัดเจน สามารถใช้ดินสอเขียนคำตอบได้
4. ห้ามการหยิบยืมสิ่งใดๆ ทั้งสิ้น จากผู้สอบอื่นๆ เว้นแต่ผู้คุมสอบจะหยิบยืมให้
5. ห้ามนำส่วนใดส่วนหนึ่งของข้อสอบออกจากห้องสอบ ข้อสอบเป็นทรัพย์สินของราชการซึ่งผู้ลักพาอาจมีโทษทางคดีอาญา
6. ผู้ที่ประสงค์จะออกจากห้องสอบก่อนหมดเวลาสอบ แต่ต้องไม่น้อยกว่า 45 นาที
7. เมื่อหมดเวลาสอบ ผู้เข้าสอบต้องหยุดการเขียนใดๆ ทั้งสิ้น
8. ผู้ที่ปฏิบัติเข้าข่ายทุจริตในการสอบ ตามประกาศคณะกรรมการการศึกษามิโทเซ คือ ได้รับ สัญลักษณ์ F ในรายวิชาที่ทุจริต และพักการศึกษาอย่างน้อย 1 ภาคการศึกษา

รับทราบ

ลงชื่อนิสิต (.....)

หมายเหตุ (เพิ่มเติม)

1. สำหรับข้อที่ให้ออกแบบ หรือ เขียนโปรแกรม คะแนนที่ได้จะแปรตามประสิทธิภาพในการทำงานของโปรแกรม
2. สำหรับข้อที่วิเคราะห์เวลาการทำงาน คะแนนที่ได้จะแปรตามความใกล้เคียงความเป็นจริงของการวิเคราะห์
3. นิสิตสามารถอ้างถึงและเรียกใช้คลาสต่าง ๆ ที่อยู่ในเอกสารประกอบการสอนได้โดยไม่ต้องเขียนขึ้นมาใหม่
4. ในข้อที่ต้องออกแบบโครงสร้างข้อมูล นิสิตไม่จำเป็นต้องตรวจสอบถึงกรณีที่มีการใส่ข้อมูลเข้าไปมากกว่าเนื้อที่ที่มีอยู่ (เสมือนว่าการจองพื้นที่นั้นจองมากพอเสมอ) หรือ กรณีที่เอาข้อมูลออกเมื่อไม่มีข้อมูลอยู่ในโครงสร้างข้อมูล
5. ให้เขียนคำตอบลงในเฉพาะพื้นที่ที่เว้นว่างไว้
6. ให้นิสิตเขียนรหัสประจำตัวและเลขที่ใน CR58 ในทุกหน้าของกระดาษคำถามด้วย



ในวิชานี้ อัตราส่วนของคะแนน quiz อยู่ที่ 15% และข้อสอบปลายภาคอยู่ที่ 45% จึงปรับอัตราส่วนของคะแนน Quiz ตามที่นิสิตต้องการ โดยนิสิตสามารถเลือกปรับน้ำหนักของคะแนน quiz ได้ตั้งแต่ 0% ถึง 15% คำนำน้หนักในส่วนที่ขาดหายไปนั้นจะถูกนำไปคิดเป็นน้ำหนักของข้อสอบปลายภาคแทน จงระบุอัตราส่วนของคะแนน quiz ที่ต้องการ โดยใส่เป็นตัวเลขจำนวนเต็มตั้งแต่ 0 ถึง 15 ลงในช่องด้านล่างนี้ต่อไป (ถ้าไม่ระบุหรือระบุค่าที่ไม่ถูกต้องจะถือว่าระบุเลข 15 ไว้) (ตัวอย่างการคิดคะแนน ถ้ากรอกเลข 4 หมายความว่านิสิตต้องการให้ quiz มีอัตราส่วนที่ 4% และสอบปลายภาคอยู่ที่ 56%)

1. (5 คะแนน) จงวาดฮัฟแมนทรี จากข้อมูลตัวอักษรในไฟล์ต่อไปนี้  
 a มีความถี่ในไฟล์ = 100, b มีความถี่ในไฟล์ = 600, c มีความถี่ในไฟล์ = 400, d มีความถี่ในไฟล์ = 300, e มีความถี่ในไฟล์ = 600

2. (5 คะแนน) ฟังก์ชัน `insertBST(t, x)` ข้างล่างนี้ เพิ่ม `x` ใน `t` ที่เป็น binary search tree โดยมีผลหลังการเพิ่มที่แตกต่างจากที่นำเสนอในชั้นเรียน อยากทราบว่า ถ้าเริ่มจากต้นไม้ว่าง แล้วเพิ่มข้อมูล 4, 5, 1 และ 9 ตามลำดับ จะได้ต้นไม้หลังการเพิ่มข้อมูลแต่ละตัวอย่างไร

```

BNode rotLC(BNode p) {
    BNode x = p->left; p->left = x->right; x->right = p;
    return x;
}
BNode rotRC(BNode p) {
    BNode x = p->right; p->right = x->left; x->left = p;
    return x;
}
BNode _insertBST(BST t, BNode p, DType x) {
    if (p == NULL) {
        t->size++;
        return newBNode(x, NULL, NULL);
    }
    if (cmp(x, p->data) < 0) {
        p->left = _insertBST(t, p->left, x); p = rotLC(p);
    } else {
        p->right = _insertBST(t, p->right, x); p = rotRC(p);
    }
    return p;
}
BNode insertBST(BST t, DType x) {
    t->root = _insertBST(t, t->root, x);
}
    
```

4			
หลังเพิ่ม 4	หลังเพิ่ม 5	หลังเพิ่ม 1	หลังเพิ่ม 9

3. สมมติว่าคุณมี hash table ที่มีขนาด  $M = 11$  ช่อง และคุณมี hash function ดังนี้

```
int h1 (int key) {
    int x;
    x = (key + 3) * (key + 5);
    x = x/15;
    x = x + key;
    x = x % 11;
    return x;
}
```

3.1 (5 คะแนน) สมมติว่าคุณแก้ไข collision ด้วย linear probe, โดยที่คุณจะ probe ไปที่ ช่องที่  $(h1(key) + i) \pmod{11}$  ถ้าหากเราเริ่มจาก hash table ที่ไม่มีข้อมูลเลย และทำการใส่ข้อมูลที่มี key ต่อไปนี้ตามลำดับ {43, 23, 1, 0, 15, 31, 4, 7, 11, 3} จงเติมค่าในตารางเพื่อระบุถึงการเก็บข้อมูลของ hash table นี้

ช่อง	0	1	2	3	4	5	6	7	8	9	10
ค่าที่เก็บ											

3.2 (5 คะแนน) ทำอีกครั้งด้วย Quadratic Probe  $(h1(key) + i^2) \pmod{11}$

ช่อง	0	1	2	3	4	5	6	7	8	9	10
ค่าที่เก็บ											

4. จงอธิบายว่าฟังก์ชัน hash ต่อไปนี้มีข้อเสียอย่างไรโดยสมมติว่าตาราง มีจำนวน 30 ช่อง

4.1 (2 คะแนน)  $h(x) = ((x*60 + 3181)*7311 + 91) \% 30$

.....

.....

4.2 (2 คะแนน)  $h(x) = (((5171*x + 31911*x*x) \% 400) / 31) \% 30$

.....

.....

4.3 (2 คะแนน)  $h(x) = (5*x + (7 + x)*4) \% 30$

.....

.....

5. (3 คะแนน) จงอธิบายว่า load factor ของ hash table คืออะไร คำนวณอย่างไร บอกอะไรเราได้บ้าง

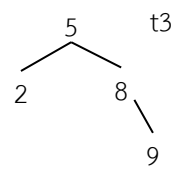
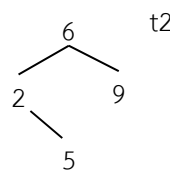
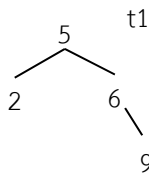
.....

.....

.....

6. (10 คะแนน) จงเขียนฟังก์ชัน DType removeMinBHeap(BHeap h) ให้ครบถ้วนสมบูรณ์ โดยฟังก์ชันนี้ทำการเอาค่าน้อยที่สุดออกจาก min heap h ไม่จำเป็นต้องคำนึงถึงกรณีที่ heap ไม่มีข้อมูล (นิสิตสามารถเขียนฟังก์ชันอื่น ๆ เพิ่มเติมได้ตามต้องการ)

7. (10 คะแนน) จงเขียนฟังก์ชันภาษาซี int equalBST(BST t1, BST t2) ที่คืน 1 เมื่อ ต้นไม้ t1 และ t2 มีข้อมูลทั้งหมดเหมือนกัน (แต่ไม่จำเป็นต้องมีโครงสร้างเหมือนกัน) ไม่เช่นนั้นให้คืน 0 กำหนดให้ t1 และ t2 มี DType เป็น float ตัวอย่างเช่น พิจารณาต้นไม้ข้างล่างนี้ จะได้ว่า t1 เท่ากับ t2 แต่ t1 ไม่เท่ากับ t3 **หมายเหตุ:** การเรียกใช้งาน equalBST นั้น เมื่อทำงานเสร็จแล้วข้อมูลใน BST ทั้งสองจะต้องคงอยู่เหมือนเดิมก่อนการเรียกใช้งาน และสามารถเขียนฟังก์ชันอื่น ๆ เพิ่มเติมได้ตามต้องการ)



8. (10 คะแนน) ให้  $t$  คือ AVL tree ที่เก็บจำนวนเต็ม จงเขียนฟังก์ชันภาษาซี `void negateAll( AVL t )` ที่คูณข้อมูลของทุกปมของ  $t$  ด้วย  $-1$  ทำให้ข้อมูลเปลี่ยน (เช่น จาก 5 เป็น  $-5$ ) และโครงสร้างของต้นไม้ก็อาจเปลี่ยนด้วย เพื่อรักษาความเป็น AVL tree โดยมีข้อกำหนดในการเปลี่ยนโครงสร้างว่า ห้ามสร้างปมใหม่ ให้เปลี่ยนได้เฉพาะ `data`, `left` และ `right` ของปมต่าง ๆ เท่านั้น (นิสิตสามารถเขียนฟังก์ชันอื่น ๆ เพิ่มเติมได้ตามต้องการ)

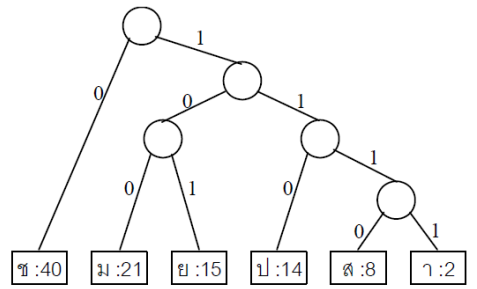
9. (10 คะแนน) จงเขียนฟังก์ชันภาษาซี `void dezip( DLinkedList a, DLinkedList b )` โดยที่  $a$  และ  $b$  เป็น circular doubly linked list with header และ  $a$  มีข้อมูลอยู่เป็นจำนวน  $2k$  ตัว ในขณะที่  $b$  มีข้อมูลอยู่ 0 ตัว หลังจากเรียกฟังก์ชันนี้ ข้อมูลตัวที่ 1, 3, 5, 7, ...,  $2k-1$  ของ  $a$  จะถูกย้ายไปเป็นข้อมูลตัวที่ 1, 2, 3, 4, ...,  $k-1$  ของ  $b$  ในขณะที่ข้อมูลตัวที่ 0, 2, 4, 6, ...,  $2k-2$  ยังคงอยู่เป็นข้อมูลตัวที่ 1, 2, 3, ...,  $k-1$  ของ  $a$  ห้ามทำการ `malloc DLinkedListNode` หรือเรียกใช้ฟังก์ชันอื่นใดที่มีการ `malloc DLinkedListNode` เด็ดขาด แต่สามารถประกาศตัวแปรประเภท `DLinkedListNode` ได้ และสามารถเขียนฟังก์ชันอื่น ๆ เพิ่มเติมได้ตามต้องการ

10. (10 คะแนน) จงเขียนฟังก์ชัน void decodeHuffman(HuffmanNode root, int[] in, int numIn) เพื่อถอดรหัสข้อมูลที่ถูกรหัสด้วยวิธี Huffman Coding โดยฟังก์ชันนี้รับ root เป็นรากของต้นไม้ in เป็นอาร์เรย์ของ 0 และ 1 ความยาว numIn ตัวที่เก็บรหัสอยู่ และฟังก์ชันนี้จะต้องเขียนข้อความที่ถอดรหัสออกมาได้ออกมาทางหน้าจอ (ด้วย printf โดยการจะ printf char นั้นจะใช้ %c) โดยรหัส 0 แทนการไปทางซ้าย รหัส 1 แทนการไปทางขวา (นิสิตสามารถเขียนฟังก์ชันอื่น ๆ เพิ่มเติมได้ตามต้องการ)

```

struct SHuffmanNode;
typedef struct SHuffmanNode *HuffmanNode;
struct SHuffmanNode {
    char    c;
    int    freq;
    HuffmanNode left;
    HuffmanNode right;
};
void decodeHuffman(HuffmanNode root, int[] in, int numIn) {

```



ตัวอย่าง 111010001111101  
 ถอดรหัสได้เป็น “สมชาย”

11. (15 คะแนน) จงออกแบบโครงสร้างข้อมูลสำหรับระบบซื้อขายหุ้นในหนึ่งวัน โดยมีรายละเอียดดังต่อไปนี้ ระบบซื้อขายหุ้นนั้นเป็นระบบที่ทำหน้าที่ “จับคู่” “คำสั่งซื้อหุ้น” เข้ากับ “คำสั่งขายหุ้น” การจับคู่คำสั่งหมายถึงการซื้อขายที่สำเร็จ คำสั่งแต่ละคำสั่งประกอบด้วยข้อมูลสามอย่าง คือ ชื่อของหุ้น, ราคา และ เวลาที่ระบบได้รับคำสั่ง คำสั่งจะมาในเวลาต่าง ๆ กัน (รับประกันว่าไม่มีคำสั่งซื้อและขายใด ๆ ที่เข้าสู่ระบบในเวลาเดียวกัน) ระบบจะได้รับคำสั่งซื้อขายเรียงตามเวลาตั้งแต่เริ่มทำการจนถึงปิดทำการ การจับคู่มีเงื่อนไขดังต่อไปนี้

- การจับคู่นั้นจะเกิดขึ้นกับคำสั่งซื้อและขายที่มี “ชื่อของหุ้น” และ “ราคา” ตรงกันเท่านั้น
- คำสั่งซื้อขายหุ้นที่ถูกจับคู่แล้วจะไม่สามารถจับคู่กับคำสั่งอื่นได้อีก
- เมื่อเราได้รับคำสั่งซื้อ(ขาย)หุ้น ณ เวลา t1 นั้น คำสั่งดังกล่าวจะจับคู่กับคำสั่งขาย(ซื้อ) **ที่ยังไม่ถูกจับคู่ ที่ได้รับมาก่อนหน้าที่สุด** (คำสั่งซื้อขายที่มีค่าเวลา t2 น้อยสุด และต้องมี “ชื่อหุ้น” และ “ราคา” ตรงกันด้วย) เราเรียกว่าการจับคู่นี้เกิดขึ้น ณ เวลา t2 (เวลาของคำสั่งขาย(ซื้อ) ที่ได้รับมาก่อน)

เราต้องการทราบราคาหุ้นแต่ละตัว ณ เวลาต่าง ๆ กำหนดให้ราคาหุ้นที่เวลา t มีค่าเท่ากับราคาของการจับคู่ของหุ้นดังกล่าว ณ เวลา t มากที่สุดที่ไม่เกิน t ตัวอย่างเช่น ถ้าไม่มีการจับคู่ดังกล่าว ให้ถือว่าราคาเป็น -1 สมมติว่าหุ้น “GUSTA” มีการจับคู่ ณ เวลา 100 ที่ราคา 50 และ ณ เวลา 200 ที่ราคา 70 เราจะถือว่าหุ้น “GUSTA” ณ t = 199 มีค่า 50, ณ t = 200 มีค่า 70 และ ณ t = 201 มีค่า -1

จงออกแบบโครงสร้างข้อมูล Market โดยกำหนดให้ชื่อของหุ้นเป็นจำนวนเต็มบวกที่มีค่าไม่เกิน 9,999 และให้ราคาของหุ้นเป็นจำนวนเต็มบวกไม่เกิน 999 และให้เวลามีหน่วยเป็น millisecond โดยเริ่มนับจากเที่ยงคืนเป็น millisecond ที่ 0 โครงสร้างข้อมูลนี้จะต้องมีฟังก์ชันดังต่อไปนี้

1. Market newMarket(int n, int \*sname) เป็นการสร้างโครงสร้างข้อมูลนี้ โดยกำหนดให้มีหุ้นอยู่ n ตัว โดยชื่อของหุ้นตัวที่ i อยู่ในตัวแปร sname[i]
2. void addBuy(Market m, int sname, int value, int t) เป็นการบอกคำสั่งซื้อหุ้นชื่อ sname ด้วยราคา value ณ เวลา t
3. void addSell(Market m, int sname, int value, int t) เป็นการบอกคำสั่งขายหุ้นชื่อ sname ด้วยราคา value ณ เวลา t
4. int getPrice(Market m, int sname, int t) เป็นฟังก์ชันที่จะคืนค่าราคาของหุ้น sname ณ เวลา t ในกรณีที่ยังไม่มี การจองหุ้นตามเงื่อนไขข้างต้น ฟังก์ชันนี้จะคืนค่า -1

ให้นักออกแบบโครงสร้างข้อมูล โดยนิสิตจะต้องเขียน struct สำหรับโครงสร้างข้อมูลนี้ และเขียนฟังก์ชันทั้ง 4 ตามโจทย์ นิสิตสามารถใช้โครงสร้างข้อมูลที่เราเรียนมาแล้วได้ทุกอย่าง และสามารถปรับเปลี่ยน DType ของแต่ละโครงสร้างข้อมูลได้ และให้ระบุด้วยโดยให้ระบุเพียงแค่ว่าใช้โครงสร้างข้อมูลอะไรบ้าง ในกรณีที่มีการเปลี่ยนแปลงฟังก์ชันต่าง ๆ เกี่ยวกับ DType (เช่น cmp, newDType) ให้เขียนโปรแกรมในส่วนฟังก์ชันดังกล่าวด้วย (หมายเหตุ: ให้เขียนคำตอบของข้อ 11 ในหน้า 7 เท่านั้น ใช้ได้ทั้งด้านหน้าและด้านหลัง)

รายการโครงสร้างข้อมูลที่ต้องการใช้ พร้อม DType ของโครงสร้างข้อมูลนั้น

.....

.....