40	
เลขที่นั่งสอบ	

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้	•
ข้อสอบปลายภาคการศึกษาที่ 2/ วันจันทร์ที่ 12 พฤษภาคม 2557 วิชา CPE 113 Algorithms and Data Structures.	2556 เวลา 9.00 - 12.00 น. น.ศ. วศ.คอมพิวเตอร์ชั้นปีที่ 1A,B
 คำสั่ง บ้อสอบมีทั้งสิ้น 6 ข้อ จำนวน 8 แผ่น(รวมแผ่นนี้) ข้อ 30 คะแนน ทำทุกข้อลงในข้อสอบที่เว้นช่องไว้ให้ ไม่อนุญาดให้นำเครื่องคำนวณใด ๆเข้าห้องสอบ ห้ามนำเอกสารใด ๆ เข้าห้องสอบ เขียนชื่อ และ รหัสประจำดัว ลงในกระดาษคำตอบท 	
	บน์ ศุภิศิริสันด์) อกข้อสอบ

ชื่อ	รหัสประจำตัว	เลขที่นั่งสอบ
1. พิจารณา คลาส Node, Lini	kNode และ Queue ซึ่งถูกเขียนไว้แล้ว ทำห	เน้าที่จัดการโหนดด้วย
โครงสร้างแบบ Queue (แสดงข้	้อมูลให้เห็นเท่าที่จำเป็น)	
class Node { int info;	-	
int height, b	alance;	
Node left, rigi	ht;	
}		
class LinkNode { Node addr;		
LinkNode n	next;	
public LinkNode (Noc	de ptr) { }	
}		
class Queue { LinkNode first	eq, lastq;	
public Queue () {	}	
void addq(Node bNo	de) { }	
Node deleteq() {	}	
}	·	
และมีโครงสร้างแบบ binary tre	e ถูกเขียนขึ้นดังนี้	
class binaryTree { Node ro	-	
public binaryTree () {	}	
Node breadthFirstSear	rch(Node root, int key) {}	
}		
ถ้าด้องการค้นหาข้อมูลในดันไม้	์ดันนี้ แบบ Breadth first search โดยด้องก	ารให้ return โหนดที่กันเจอ
_	จงเดิมคำสั่งให้ method breadthFirstSearcl	
Node breadthFirstSearch (No	de root, int key)	
{ Node current = root;		
		}

ชื่อ	รหัสประจำตัว	เลขที่นั่งสอบ
~ f	مرين والمراجع المراجع	
2. จากโครงสร้างโหนดในข้อ 1 สมมุ		
ในเมธอด AVLtype เพื่อที่ทำหน้าที่ค	รวจสอบการสมคุล ของ balance f	factor แล้ว return หมายเลขดาม
ลักษณะการไม่สมคุลที่เกิดขึ้น คือ		
0 กรณี จันไม้สมดุล		
1 กรณี ดันไม้ไม่สมดุลแบบ	left of left	
2 กรณี ดันไม้ไม่สมดุลแบบ		
3 กรณี ดันไม้ไม่สมดุลแบบ		
ง 4 กรณี ตันไม้ไม่สมดุลแบบ		
int AVLtype(Node n) {		
int Avetype(Node ii) (
		,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
	,	

	<i>.</i>	
}		

ข้อสอบวิชา CPE 113 Algorithms and Data Structures ปลายภาคการศึกษาที่ 2/2556 หน้าที่ 3

3) ในการเขียนโปรแกรมเรียงลำดับข้อมูลในอาร์เรย์ แบบ Heap มีการสร้าง class ดังนี้ class DataArray { int [] data;

int count:

}

data เป็นตัวแปรอาร์เรย์ที่ใช้เก็บข้อมูล เริ่มเก็บดั้งแต่ดัวที่ 0 จนถึง count-1 ในส่วนของเมชอด heapSort ที่กำหนดให้ เป็นปรับปรุงการเรียงลำดับ ให้สามารถเรียงลำดับข้อมูลใน data[] เริ่มตั้งแต่ตัวที่ 0 จนถึง count-1 (จำนวน count ตัว) ต้องการสร้าง method siftDown () เพื่อปรับปรุงข้อมูลในอาร์เรย์ ตั้งแต่ตัวที่ 0 จนถึงตัวที่ n ให้เป็น แบบ Heap โดยให้ทำงานสอดคล้องกับเมชอด heapSort ที่เขียนขึ้น คำแนะนำ

- 1. โจทย์ข้อนี้เป็นการเรียงลำดับข้อมูลดั้งแค่ตัวที่ data[0] ถึง data[count-1]
- 2. Heap Sort โดยทั่วไปจะเก็บข้อมูลอยู่ในลำดับที่ data[1] ถึง data[count] ดังนั้นจะต้องมีการ ชดเชย โดยการลดหรือเพิ่มหมายเลขตำแหน่งของโหนดที่เกี่ยวข้องกับการเปรียบเทียบให้ ถูกด้อง
- 3. เมชอด siftDown ที่ด้องการให้สร้าง จะด้องชดเชยตำแหน่งของโหนตให้ถูกต้อง และ สอดคล้องกับการทำงานของเมชอด heapSort

ชื่อ	••••••	•••••	•••••					รหัธ	กประจำ	าตัว		•••••	เลขา	านั่งสอบ
public vo	id si	ftDo	wn(i	int i,	int n)								
{	•••••					••••		·····		····			· • • • • • • • • • • • • • • • • • • •	·····
														••••••
														••••••
														••••••
														•••••
														••••••
••••••	••••	• • • • • • • • • • • • • • • • • • • •		••••	•••••		••••	•••••	•••••	••••••	•••••	•••••		
••••••	•••••		•••••			• • • • • • • • • • • • • • • • • • • •	•••••	•••••	••••••	••••••				
	•••••	••••••	•••••	•••••		•••••		· · · · · · ·			• • • • • • • • • • • • • • • • • • • •		•••••	•••••
•••••	•••••	•••••		•••••	•••••	• • • • • • •	•••••	•••••	•••••	••••••	•••••	•••••	••••••	•••••
•••••	•••••	•••••	•••••	•••••				• • • • • • • • • • • • • • • • • • • •	••••••		••••••	•••••	• • • • • • • • • • • • • • • • • • • •	•••••
4. จงแสด	ง <mark>รู</mark> ปเ	เค่ละ	ขั้นด	อนที่เ	.ปลี่ย	นไปใ	ในแค่	ละรอ	บเพื่	อแปลงก	เราฟที่อ	ยู่ในรูปข	อง matri	x ที่กำหนดให้
ให้กลายเป็	็น m	inim	um s	pann	ing t	ree	โดยใ	ชัหลั	ักการ	ของ Prir	n 's Alg	orithm		
	Α	В	С	D	E	F	G	Н	ı					
Α	0	2	3	2	0	4	0	4	0					
В	2	0	0	3	6	3	4	6	0			(H)		
С	3	0	0	7	0	2	0	8	6					
D	2	3	7	0	7	1	4	0	0		(A)		(B)	
E	0	6	0	7	0	0	5	7	5					(F)
F	4	3	2	1	0	0	2	0	3 7	(C)		\bigcirc		(E)
г G	0	3 4	0	4		2	0	0	8		Œ			
	4				5						(F)		(G)	
Н		6	8	0	7	0	0	0	0			(I)		
ł	0	0	6	0	5	7	8	0	0			•		

do	รหัสประจำตัว	เลขที่นั่งสอบ
----	--------------	---------------

ชื่อ	รหัสประจำดัว	เลขที่นั่งสอบ
5. กำหนดให้ ข้อมูลที่ด้องการนำมาเก็บมีโคร	งสร้างดังนี้	
class data {long sid;		
}		
class dataArray { data[] table ;		
}		
•	9 2 2 2 2	· · · · ·
ฟังก์ซัน Hashing คือ h(k) = ตัวเลขข้อมู		ลกรอย และหลกหนวย
เช่น 9876543210 จะถูกเ		
ฟ้งก์ชันสำหรับแก้ปัญหาการชนกันของข้อมูล	คือ h'(k) _i = h(k) + i² โดยส์	มีขนาดของ ด ารางเก็บข้อมูล
10000 ช่อง ซึ่งคำแหน่งที่ไม่มีข้อมูลเก็บอยู่ค่	า sid ในคำแหน่งนั้นจะเป็น	. 0
จงสร้างฟังก์ชันที่ใช้ในการคันหาดำแหน่งที่เก็	้บของข้อมูลโดยการคันหาจ	จากฟิลด์ sid ถ้าคันเจอให้
retum ดำแหน่งที่เก็บอยู่ ถ้าคันไม่เจอให้ ret	Tu .	
v		
int Search_Hashing (long int key)		
{		
		••••••
		•••••••••••••••••••••••••••••••••••••••
	***************************************	•••••••••••••••••••••••••••••••••••••••
		••••••••••••
	••••••	
1		

ชื่อ			•			รหัสป	ระจำตัว		•••••	เล	ขที่นั่งสอบ	
6.	. จงแสดงรูปดันไม้ของ Huffman ที่เกิดจากการเข้ารหัสของข้อมูลต่อไปนี้											
	นามีรูงู่⊟นามีรูปู่⊟มานีพาตามานา⊟มานีพาดาดูปู											
จา	จากดันไม้ที่ใต้ จงกำหนตรหัสที่ใช้ย่อขนาดของแต่ละดัวอักษร											
		v	ମ	ମ	и	Л	W	ม	3	า	ే	្ន
-												
		بو	م. ملد		.1							
จ้า	จำนวนบิททั้งหมดที่ใช้ในการย่อข้อความที่กำหนดให้ บิท											