



มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี  
การสอบปลายภาคเรียนที่ 2 ปีการศึกษา 2559

วิชา CPE 111 Programming with Data Structures.

วศ.คอมพิวเตอร์ ปีที่ 1 กลุ่มที่ 1, 2

สอบวันพุธที่ 17 พฤษภาคม 2560

เวลา 09.00 -12.00 น.

คำชี้แจง

1. ข้อสอบแบ่งเป็น 7 ข้อ จำนวน 8 แผ่น ประกอบด้วยใบปะหน้าข้อสอบ 1 แผ่น ตัวข้อสอบ 7 แผ่น (หน้า 2 -8) มีคะแนนรวม 35 คะแนน
2. เขียนโปรแกรมเพื่อหาคำตอบลงในส่วนที่กำหนดไว้ท้ายข้อโดยใช้ภาษาจาวา
3. อนุญาตให้นำเครื่องคำนวณตามระเบียบของมหาวิทยาลัยเข้าห้องสอบได้
4. ไม่อนุญาตให้นำเอกสาร หรือกระดาษจดบันทึกใดๆ เข้าห้องสอบ
5. เขียนชื่อ และ รหัสประจำตัว ลงในกระดาษคำตอบทุกแผ่น (และแผ่นนี้)

ชื่อ-สกุล..... รหัสนักศึกษา.....

เมื่อนักศึกษาทำข้อสอบเสร็จ ต้องยกมือบอกกรรมการคุมสอบ

เพื่อขออนุญาตออกนอกห้องสอบ

ห้ามนักศึกษานำข้อสอบและกระดาษคำตอบออกนอกห้องสอบ

นักศึกษาซึ่งทุจริตในห้องสอบ อาจถูกพิจารณาโทษสูงสุดให้พ้นสภาพการเป็นนักศึกษา

(อ.พิพัฒน์ ศุภศิริสันต์ )

ผู้ออกข้อสอบ (086 770 7033)

ข้อสอบนี้ได้ผ่านการประเมินจากคณะกรรมการภาควิชาวิศวกรรมคอมพิวเตอร์

(รศ.ดร.พีรพล ศิริพงศ์วุฒิก)

ประธานหลักสูตร

วันที่ ..... เดือน ..... พศ .....

กำหนดให้

คำสั่งเกี่ยวกับการสร้างคลาส Implements Comparator Comparable compare compareTo

คำสั่งเกี่ยวกับสตริง compareTo(), compareToIgnoreCase(), equals(), equalsIgnoreCase(), length(), toLowerCase(), toUpperCase()

คำสั่งเกี่ยวกับอาร์เรย์ลิสต์    `add ()`, `addAll()`, `get()`, `contains()`, `indexOf()`, `remove()`, `size()`

คำสั่งเกี่ยวกับแฮชแมป put(), get(), containsKey(), containsValue(), remove(), size(), KeySet()

## คำสั่งเกี่ยวกับ Collecitons Collections.sort(), Collections.binarySearch()

1. ถ้าต้องการออกแบบไหนดทที่มีคุณสมบัติเป็น **Binary Tree** โดยมีข้อมูลที่ต้องการเก็บดังนี้

รหัสไหนดี เป็นตัวอักษร

ความถี่ เป็นตัวเลขจำนวนเต็ม

ซึ่งโหนดที่ออกแบบอาจถูกนำไปใช้ในการเปรียบเทียบโดยใช้ความถี่

จงออกแบบคลาสชื่อ **Node** เพื่อเก็บข้อมูลในลักษณะของโบนารีทรี (มีโหนดลูกทางซ้ายและขวา) และมีการ implements Comparable ในการเปรียบเทียบโดยใช้ความถี่ (3 คะแนน)

This image shows a full page of white paper with horizontal dashed lines, typical of primary school writing paper. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.







ชื่อ ..... รหัสประจำตัว.....เลขที่นั่งสอบ.....

5. ถ้าต้องการบีบอัดข้อมูลคำว่า "CPE 30 YEARS ANNIVERSARY" (รวม 24 ตัวอักษร) โดยมีการนับและใส่ตัวอักษรเริ่มต้นเข้าไปใน Priority Queue เป็นตามลำดับ ดังนี้

□	0	3	A	C	E	I	N	P	R	S	V	Y
3	1	1	3	1	3	1	2	1	3	2	1	2

ในการจับคู่เพื่อสร้างโหนดในต้นไม้ ถ้าอัลกอริทึมที่ใช้จับคู่จะดึงข้อมูลตัวแรกจากคิว มาสร้างเป็นโหนดทางซ้าย และดึงตัวที่สองมาสร้างเป็นโหนดทางขวา จงแสดงรูปต้นไม้ของ Huffman ที่เกิดจากข้อมูลชุดนี้ (ลำดับการจับคู่ต้องเป็นไปตามการทำงานของ Priority Queue ที่กำหนดไว้)

กำหนดให้การทำงานของ Priority Queue ที่ใช้ ถ้าข้อมูลที่เข้าใหม่ มีลำดับความสำคัญเท่ากับที่มีอยู่ จะนำมาต่อท้าย แต่ถ้ามีความสำคัญมากกว่าจะนำไปอยู่ข้างหน้า (7 คะแนน)

จากต้นไม้ที่ได้ จงกำหนดรหัสที่ใช้ย่อขนาดของแต่ละตัวอักษร

□	0	3	A	C	E	I	N	P	R	S	V	Y

จำนวนบิตรวมที่เกิดขึ้นทั้งหมดของข้อมูลชุดนี้คือ ..... บิต

Binary Sequence (8-bits) ของข้อมูลชุดนี้ แทนได้ด้วยเลขฐาน 16 คือ

.....

ชื่อ ..... รหัสประจำตัว ..... เลขที่นั่งสอบ .....

6. จงแสดงข้อมูลในตาราง Hashing ที่เกิดขึ้นเมื่อนำข้อมูลเหล่านี้มาเก็บในตารางตามลำดับ(4 คะแนน)

38, 64, 76, 115, 129, 141, 157, 195, 375

กำหนดให้ ฟังก์ชันที่ใช้เก็บข้อมูล และ แก้ปัญหาการชนกันคือ  $h(k) = k \% 13$

ฟังก์ชันที่ใช้ในการแก้ปัญหาเมื่อเกิดการชนคือ  $i^2$

0	1	2	3	4	5	6	7	8	9	10	11	12

7. กำหนดให้ กราฟแบบไม่มีทิศทางระหว่างโหนด A - L มีการแสดงคุณสมบัติโดยใช้ Adjacency matrix ดังนี้

	A	B	C	D	E	F	G	H	I	J	K	L
A		9	8	4		4						
B	9			8	10							
C	8					7		7				
D	4	8			4	2	4		3			
E		10		4			4			6		
F	4		7	2				8	1		2	
G				4	4				3	5		
H			7			8					8	
I				3		1	3			6	1	6
J					6		5		6			6
K						2		8	1			6
L									6	6	6	

ถ้าต้องการแปลงกราฟนี้ให้เป็น minimum spanning tree โดยใช้อัลกอริทึมของ Prim และใช้ priority queue ช่วยในการจัดลำดับการเลือกเอดจ์ โดยที่การดึงข้อมูลจากกราฟใส่ลงใน queue ให้เลือกเอดจ์ตามลำดับจากซ้ายไปขวา และบนลงล่างของเมตริกซ์ จงแสดงลำดับของเอดจ์ที่ถูกเลือก และ minimum spanning tree ที่เกิดขึ้น (5 คะแนน)

ลำดับของเอดจ์ที่ถูกเลือก

--	--	--	--	--	--	--	--	--	--	--	--	--

minimum spanning tree ที่เกิดขึ้น

