Name……id ……..……section…….CR58……

FACULTY OF ENGINEERING

CHULALONGKORN UNIVERSITY

2190221 Fundamental Data Structure and Algorithm

Year 2, Second Semester, Final Examination Date 6 May 2015 Time 08.30-11.30 (3 hours)

Important

1. This exam paper has 8 questions. There are 5 pages in total including this page. The total mark is 46.
2. Write your answers in the provided answer book.
3. Write your name and ID on every page of this exam paper.
4. When the exam finishes, students must stop writing and remain in their seats until all question sheets are collected and the examiners allow students to leave the exam room.
5. A student must sit at his/her desk for at least 45 minutes.
6. A student who wants to leave the exam room early (must follow (5)) must raise his/her hand and wait for the examiner to collect his/her papers. The student must do this in a quiet manner.
7. No books, lecture notes or written notes of any kinds are allowed in the exam room.
8. No calculators are allowed.
9. A student must not borrow any item from another student in the exam room. If you want to borrow an item, ask the examiner to do it for you.
10. Do not take any part of the question sheet and answer books out of the exam room. All papers are properties of the government of Thailand. Violators of this rule will be prosecuted in a criminal court.
11. A student who violates rules will be considered as a cheater and will be punished by the following rule:
    - **Suspected cheaters will get an F in the subject they are suspected to cheat, and will not be able to enroll in the next semester.**
    - **Cheaters will get an F in the subject they are caught cheating, and will not be able to enroll in the next 2 semesters.**

I understand and agree to the given instructions.

Signature(……)

1. (5 marks) Given a hash table for integer data of size 13. Let *hash(x) = x%TableSize* and $hash_2(x) = 7-(x\%7)$. **Show and explain** (step by step) what happens when 1, 5, 18, 8 are inserted into the table in order, using double hashing.
2. (4 marks) In open addressing hash table, explain the reason for the use of lazy deletion. Give example(s) too.
3. (6 marks) A Binary search tree storing integer values is constructed using AVLNode, which has the following code:

```
public class AVLNode{
        public int value;
        public int height; //height of the tree, counting from this node to leaf.
        public AVLNode left;
        public AVLNode right;

}
```

Assume that the value of the height is always correctly updated and the tree is always a binary search tree, write code for method:

public boolean isAVL(AVLNode n)

This method tests to see whether node n and all nodes below it satisfy the structural requirement of AVL tree.

4. (6 marks) The code for Binary Search tree of non-duplicated integers is given below.

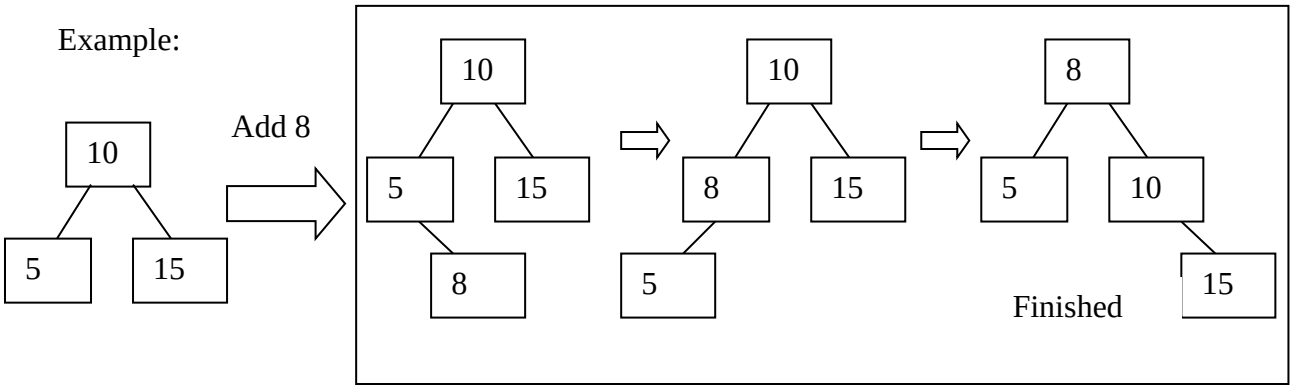
```
class BSTNode{
        int data;
        BSTNode left;
        BSTNode right;
       BSTNode parent;

        public BSTNode(int value, BSTNode l, BSTNode r, BSTNode p){
                data = value;
                left = l;
                right = r;
              parent =p;
        }
}

class BST{
        BSTNode root; int size;

        //Let the subtree we consider has n as its root,
        //          this method rotates the left child of n up the subtree.
        //Return the changed root of the subtree.
        public BSTNode rotateLeftChild(BSTNode n){… } // can be called directly.

        public BSTNode rotateRightChild(BSTNode n){… } // can be called directly.
}
```

write method:

public BSTNode addUP(int num, BSTNode n)

This method adds **num** to part of the tree that has **n** as its root. After num is added, it must be at the root of that part of the tree (use rotation to move a newly added number up the tree).

Example:



5. (5 marks) Write the code for the following method:

public static int[] selectionsort(int[] input)

This method performs selection sort by keep finding the minimum number and swapping it to its correct position in the array one by one. Assume that there is method:

public static void swap(int index1, int index2, int[] a) that you can use to swap values in position index1 and index2 of array a.

6. (4 marks) If you want to use heap to compress a text file that contains the following alphabets, with the following frequencies:

a – 320
b- 400
c- 530
d- 670
e- 1300

Assume that you always use the first value taken from the heap as left branch of your Huffman tree, and the second value from the heap as right branch, draw your Huffman tree for each step of removing 2 lowest values from the heap, until the Huffman tree is complete.

7. (7 marks) Class Heap (a min-heap that stores positive integers) has the following variables and methods:

public class Heap{

```
int[] mData;
int size; //number of elements in the heap

public Heap(){
        // a working constructor that initializes everything correctly.
        // Do not write code for this. You can call this constructor.
}

public void add (int element) {
        if (++size == mData.length) {
                int[ ] newHeap = new int [2 * mData.length];
                System.arraycopy (mData, 0, newHeap, 0, size);
                mData = newHeap;
        }
        mData [size - 1] = element;
        percolateUp();
}

public int pop( ) {
        if (size==0)
                throw new NoSuchElementException("Priority queue empty.");
        int minElem = mData [0];
        mData [0] = mData [size – 1];
        mData [--size] = minElem;
        percolateDown (0);
        return minElem;
}

public void percolateDown(int start) {
        int parent = start;
        int child = 2*parent+1;
        int temp;
        while (child<size) {
                if(child<size-1 && mData[child] > mData[child+1])
                        child++;
                if (mData[parent] <= mData[child])
                        break;
                temp = mData[child];
                mData[child] = mData[parent];
                mData[parent] = temp;
                parent = child;
                child= 2*parent+1;
        }
}

public void percolateUp() {
    int parent;
    int child = size-1;
    int temp;
    while (child>0) {
        parent = (child-1)/2;
        if(mData[parent] <= mData[child])
            break;
        temp = mData[parent];
        mData[parent] = mData[child];
        mData[child] = temp;
        child = parent;
    } // while
 } // method percolateUp
        …
```

}

The following method is to be added to class Heap:

- *public void removeValue(int value)*: This method removes specified value stored in array heap. The array after the removal must still have the quality of heap.

  Write the code for *removeValue(int value).* Discuss the asymptotic runtime of your code.

8. (9 marks) If you want to sort an array of positive integers containing n numbers but you **do not know the range of the values in the array**, explain and write code for method:

   public static int[] sort(int[] input)

   This method sorts the array so that the asymptotic runtime is less than O(n log n) (this means it is faster than mergesort and quicksort). Discuss the runtime of your code and any limitation your code has.