Exam
Data structures DAT036/DIT960

Time	Thursday 21st August 2014, 8:30–12:30
Place	Maskinhuset
Course responsible	Nick Smallbone, tel. 0707 183062

The exam consists of **six questions**.

For a **3** (Chalmers) or **G** (GU), you need to answer **three** questions correctly. You can ignore any parts labelled **"For a 4"** or **"For a 5/VG"**.

For a **4** (Chalmers only), you need to answer **four** questions correctly. You must also answer all parts labelled **"For a 4"** in those questions.

For a **5** (Chalmers) of **VG** (GU), you need to answer **five** questions correctly. You must also answer all parts labelled **"For a 5/VG"** in those questions.

For an answer to be considered correct, it should contain no major mistakes. Minor mistakes might be accepted, but this is at the discretion of the marker.

When a question asks for **pseudocode**, you can use a mixture of English and programming notation to describe your solution, and should give enough detail that a competent programmer could easily implement your solution.

Allowed aids One A4 piece of paper of hand-written notes, which should be handed in after the exam. You may write on both sides.

You may also bring a dictionary.

Note Begin each question on a new page.

Write your anonymous code (*not* your name) on every page.

Good luck!

1. Here is an algorithm to test if two arrays `X` and `Y` are disjoint (contain no elements in common):

```
S = new empty AVL tree
for every element x in X,
    S.insert(x)
for every element y in Y,
    if S.member(y) then
        return false
return true
```

What is the worst-case time complexity of this algorithm?

For a 3/G, you may assume that `X` and `Y` have the same length, *n*.
Give the complexity in terms of *n*.

For a 4/5/VG, don't assume that `X` and `Y` have the same length.
Give the complexity in terms of *m* and *n*, where *m* is the length of `X` and *n* is the length of `Y`.

For a 5/VG, suppose that we run the algorithm twice, with the following inputs. Which run would you expect to go faster, if either?

a) `X` is an array of 1000000 elements, `Y` is an array of 10 elements

b) `X` is an array of 10 elements, `Y` is an array of 1000000 elements

2. Design an algorithm that takes:

   - An array containing n distinct natural numbers
   - A number $k \leq n$

   and calculates the sum of the k largest numbers in the array.

   For example, if the array is {3, 7, 5, 12, 6} and $k = 3$, then the algorithm should return 25 (12+7+6).

   Write down your algorithm as **pseudocode** – you don't need to write fully detailed Java code. **You may freely use standard data structures and algorithms from the course in your solution, without explaining how they are implemented.**

   For a 3/G: your algorithm should take O(n log n) time.

   For a 4/5/VG: your algorithm should take O(n log k) time.

3. Design a data structure for storing a set of integers. It should support the following operations:
    - `new()`: create a new, empty set
    - `insert(x)`: add an integer `x` to the set
    - `member(x)`: test if a given integer `x` is in the set
    - `delete(x)`: delete an integer `x` from the set
    - `deleteLessThan(x)`: delete all numbers that are strictly less than `x` from the set

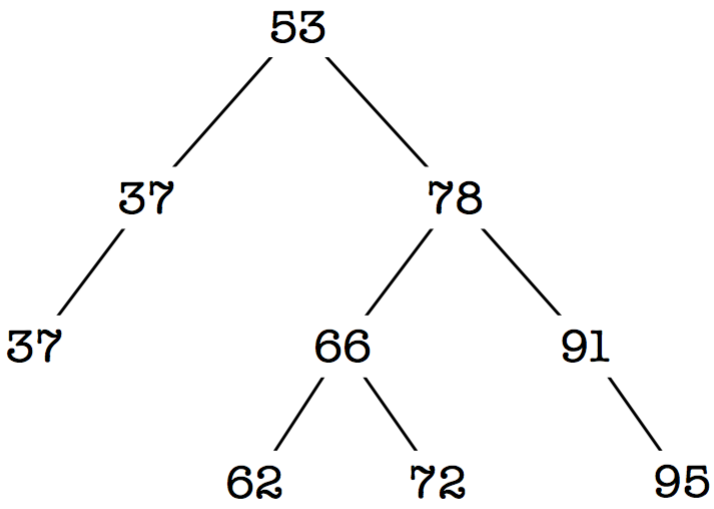
    You may freely use standard data structures and algorithms from the course in your solution, without explaining how they are implemented.

    You should say what design or existing data structure you have chosen, and give **pseudocode** for each of the operations – you don't need to write fully detailed Java code.

    The operations must have the following time complexities:

    - **For a 3/G:**
      O(1) for `new`,
      O(log n) for `insert/member/delete`,
      O(n log n) for `deleteLessThan`
      (where n is the number of elements in the set)
    - **For a 4/5/VG:**
      as for 3/G but the complexity of `deleteLessThan` must be strictly better than O(n) (e.g., O(log n)).

4. You are given the following binary search tree.



a) Colour the nodes of the tree red and black so that it becomes a valid red-black tree. If you don't have a coloured pen, you could e.g. draw a circle for red nodes and a square for black nodes.

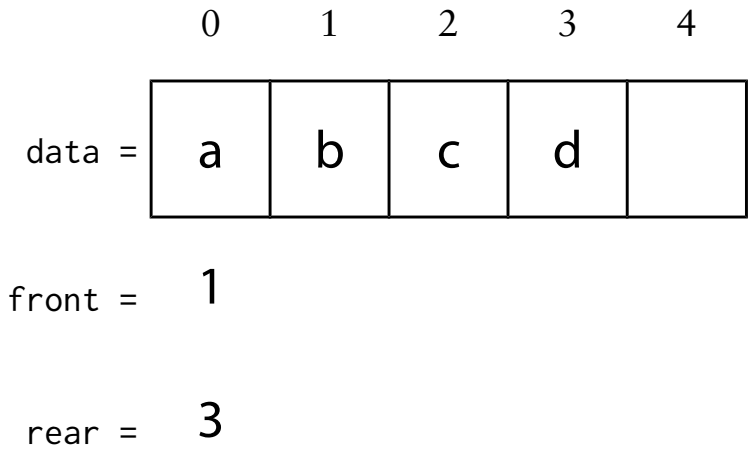
b) Insert 60 into the tree using the red-black insertion algorithm (pick either top-down or bottom-up insertion). Write down the final tree.

Erratum: DAT036 students can answer an alternative part b:
Insert 60 into the tree using the splay tree insertion algorithm. Write down the final tree.

5. Suppose we have a class `ArrayQueue` which implements a bounded queue as a circular array with the following variables and methods:

```
class ArrayQueue<E> implements Queue<E> {
  private E[] data;
  private int front, rear;
  public ArrayQueue(int capacity);
  // Add an item to the queue.
  public boolean offer(E item);
  // Remove an item from the queue.
  public E poll();
}
```

If we create a queue `q` of capacity 5, insert four elements and then remove one, then `q`'s instance variables will be as follows afterwards:



Suppose that we now execute the following sequence:

```
q.offer("e"); q.offer("f"); q.poll(); q.poll();
q.offer("g"); q.poll();
```

What will `q`'s instance variables contain afterwards?

6. Take a look at the following function:

```
void f(int[] a, int m) {
  int[] b = new int[m];
  for (int i = 0; i < a.length; i++) {
    int j = a[i];
    b[j]++;
  }
  int i = 0;
  for (int j = 0; j < m; j++) {
    for (int k = 0; k < b[j]; k++) {
      a[i] = j;
      i++;
    }
  }
}
```

a) What is the result of running the following code?

```
int[] a = {5,3,7,1,0,3,5,4,3,6};
f(a, 8);
System.out.println(Arrays.toString(a));
```

b) What does the function do?