Linköping University
Department of Science and Technology/ITN
Aida Nordman

Exam: TND004, Data structures

2015-06-01, 14.00 - 18.00

- This is an **individual open book exam**. Thus during the exam, you are allowed to use any books, solutions for labs and previous exams.
- Solutions to different exercises should be placed **one-sided on separate page(s)**, clearly identified. You should also **number all sheets** you are going to hand in and **sort them.**
- Justify properly your answers: missing or insufficient explanations will result in reduction of points.
- All answers should be given in your own words.
- Be precise and clear in your answers. The clarity of your answers is also evaluated. You may use diagrams to help in explaining your ideas.
- Unreadable text will be ignored. You can answer in Swedish or English.
- If a problem allows different solutions, e.g. algorithms of different complexity, only optimal solutions give the maximal number of points.
- Your grade is decided as the table below indicates.

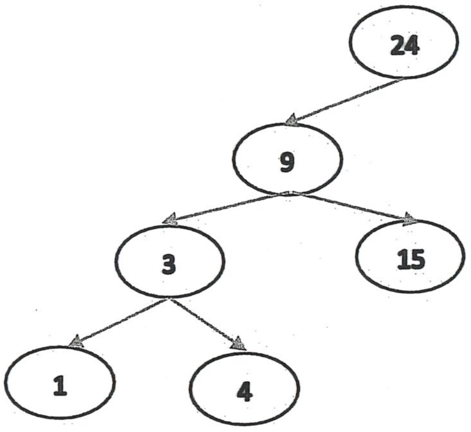
Final Grade	**Requirements**
3	At least 12 points in exam's **Part I**.
4	• At least 12 points in exam's **Part I** and • at least 10 points in **Part II** or in **Part III**.
5	• At least 12 points in exam's **Part I**, and • at least 10 points in **Part II**, and • at least 12 points in **Part III**.

Good Luck!

Part I

Exercise 1 *[6p]*

Consider the following binary search tree (BST).



a. List all the possible insertion orders (i.e. permutations) of the keys that could have produced the BST above.

b. Assume the BST above is also treated as a splay-tree. Show the tree obtained after accessing **4**.

c. Is the BST obtained in b an AVL-tree? Motivate.

Exercise 2 *[4p]*

Write the pseudo-code of a function called **`findLargest`** that finds the largest number in an array A with $n > 0$ integers, using a divide-and-conquer strategy. What is the time and space complexity of your algorithm? Motivate your answer.

Exercise 3 *[2p]*

In lab 2 of this course, it was requested that you implemented an open addressing hash table, using linear probing strategy to resolve collisions.

Explain in your own words why it is needed to mark the deleted slots of the table with a special item (in lab 2, an instance of class **`Deleted_Item`** was used).

Exercise 4 *[8p]*

Consider the following sequence, where $N > 1$ is an integer constant.

`2 1 4 3 6 5 ... N N-1`

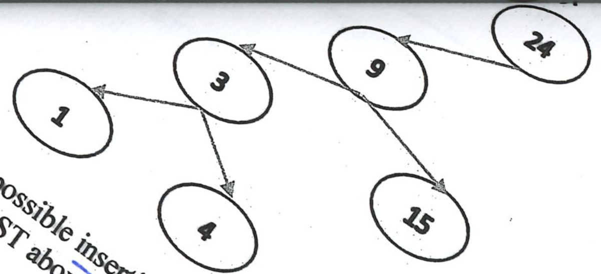
Assume you want to sort increasingly the sequence above. What is the running time of each of the algorithms below having as input the sequence above. Use big-Oh notation and motivate your answer.

a. Insertion-sort.

b. Bubble-sort.

c. Merge-sort.

d. Quicksort, using median-of-three as partition strategy.



a. List all the possible insertion orders (i.e. permutations) of the keys that could have produced the BST above.

b. Assume the BST above is also treated as a splay-tree. Show the tree obtained after accessing **4**.

c. Is the BST obtained in b an AVL-tree? Motivate.

Exercise 2 *[4p]*

Write the pseudo-code of a function called **findLargest** that finds the largest number in an array **A** with $n > 0$ integers, using a divide-and-conquer strategy. What is the time and
pace complexity of your algorithm? Motivate your answer.

cise 3 *[2p]*

of this course, it was requested that you implemented an open addressing hash table,
r probing strategy to resolve collisions.
ur own words why it is needed to mark the deleted slots of the table with a
lab 2, an instance of class **Deleted_Item** was used).

[8p]

sequence, where $N > 1$ is an integer constant.

2 1 4 3 6 5 ... N N-1

asingly the sequence above. What is the running time of each of
nput the sequence above. Use big-Oh notation and motivate

Merge-sort.

icksort, using median-of-three as partition strategy.

Part II

Exercise 5 *[10p]*

In lab 1, you implemented a function that merged two sorted lists into a single sorted list (called **_union**).

Consider that there are $k > 2$ sorted lists, each with length $n > 0$. Describe an algorithm that merges the k sorted lists into a single sorted list with length $k \times n$. Indicate clearly which data structures you would use to solve this problem and the time complexity of your solution.

Exercise 6 *[10p]*

Suppose that one of your colleagues claims to have improved the insertion sort by using binary search to find the position **p** where each new insertion should take place, as the algorithm below indicates (i.e. function **binary_search** returns a position **p** such that either **A[p-1]** ≤ **temp** < **a[p]** or, **temp** < **A[p]** and **p** = **0**).

```
void improved_insertion_sort(int A[], int n)
{
    int i, j;
    int temp;

    for (j = 1; j < n; j++)
    {
        temp = A[j];
        int p = binary_search(A, j-1, temp);

        //Move A[p],⋯, A[j-1]
        for(int i = j-1; i >= p; --i)
            A[i+1] = A[i];

        A[p] = temp; //Place temp in A[p]
    }
}
```

a. What is the worst-case complexity of **improved_insertion_sort**, if you take only account of the comparisons (between array **A**'s elements) made by **binary_search**? Motivate your answer.

b. What is the worst-case complexity of **improved_insertion_sort**, if only moves of the data values are taken into account? Motivate your answer.

c. Is your colleague right in claiming that she has improved the insertion sort algorithm? Motivate your answer.

Part III

Exercise 7 *[20p]*

Consider a transport company that owns a number of flights connecting $n > 1000$ different geographical points in the world. This company sells flight tickets to customers who want to travel from a departure point to a destination point. In the case that there is no direct flight between both points, it may be possible to take several flights (i.e. a trip can involve stops in other geographical points, before the destination). All customers flying from a point A to a point B pay the same price for the flight and, in the case of non-direct flights, the price of the trip is the sum of the cost of each flight taken from the departure to the trip's destination.

As part of the customer service, the company wants to be able to suggest automatically to its customers the cheapest trip from a given departure point to a given destination point. Moreover, if there are several possible trips with the lowest price then the one with the fewest number of stops in other geographical points (i.e. before the final destination is reached) should be suggested to the customer.

Propose a solution to solve efficiently the problem described above. Assume that the company offers about $4n$ direct flights and there is at most one direct flight between every pair of geographical points.

The description of your solution must describe

- the data structures used,
- the algorithm(s) used, and
- the time complexity of the algorithm(s).