Linköping University
Department of Science and Technology/ITN
Aida Nordman

Exam: TND004
Data structures

2015-10-20, 08.00 - 12.00

- 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, brief, 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 *[8p]*

a. Show the result of inserting the sequence $20, 17, 28, 77, 33, 80, 10, 12$ into an initially empty binary search tree.

b. Show the tree obtained after deleting the root of the tree obtained in a.

c. Is the tree obtained in b and AVL-tree? Motivate your answer.

d. Consider the following sorting algorithm, known as **tree sort**, using a binary search tree. Tree sort first builds a binary search tree from the keys to be sorted, and then traverses the nodes of the tree in inorder, so that the keys come out in sorted order.

What is the time complexity of the **tree sort** algorithm, in the worst-case? Motivate your answer and give an example of a worst-case sequence.

Exercise 2 *[6p]*

Consider the function **mystery** as defined below. The function has an argument $n \geq 0$.

```
int mystery(unsigned int n)
{
    if (n == 0) return 2;
    return mystery(n/2) * mystery((n-1)/2);
}
```

a. Give a non-recursive mathematical function $f(n)$ that describes what the **mystery** function above calculates.

b. What is the time complexity of the **mystery** function? Motivate your answer.

c. What is the space complexity of the **mystery** function? Motivate your answer.

Exercise 3 *[6p]*

a. Consider a sequence S with $n > 0$ values of which only $0 < k < n$ are unique values. Moreover, the k unique values occur contiguously in the sequence, while the other $(n-k)$ values are equal to each other. For instance, $\langle 5\ 5\ 5\ 5\ 5\ 5\ 5\ 5\ 3\ 1\ 2\ 5\ 5 \rangle$ and $\langle 3\ 1\ 2\ 5\ 5\ 5\ 5\ 5\ 5\ 5\ 5\ 5\ 5 \rangle$ are examples of such sequence S.

Indicate the time complexity of the insertion sort, in the worst-case, when sorting a sequence such as S. Express the time complexity as a function of k and n. Give also a concrete example of a worst-case sequence with the format described above. Motivate your answer.

b. Consider the increasingly sorted sequence $S = \langle 1\ 2\ 3 \ldots k\ x\ x\ x\ x\ x \ldots\ x \rangle$ with $n > 0$ values, where $0 < k < n$ and $x > k$ is a constant. Indicate the time complexity of the quicksort algorithm, when sorting S, using the first element as the pivot. Express the time complexity as a function of k and n. Motivate your answer.

, 17, 28, 77, 33, 80, 10, 12 into an initially
ng the root of the tree obtained in a.
AVL-tree? Motivate your answer.

sorting algorithm, known as **tree sort**, using a binary search
builds a binary search tree from the keys to be sorted, and then
odes of the tree in inorder, so that the keys come out in sorted order.
the time complexity of the **tree sort** algorithm, in the worst-case? Motivate
ur answer and give an example of a worst-case sequence.

ercise 2 *[6p]*

Consider the function **mystery** as defined below. The function has an argument $n \geq 0$.

```
int mystery(unsigned int n)
{
    if (n == 0) return 2;
    return mystery(n/2) * mystery((n-1)/2);
}
```

a. Give a non-recursive mathematical function $f(n)$ that describes what the **mystery** function above calculates.
What is the time complexity of the **mystery** function? Motivate your answer.
at is the space complexity of the **mystery** function? Motivate your answer.

[6p]

sequence S with $n > 0$ values of which only $0 < k < n$ are unique values.
k unique values occur contiguously in the sequence, while the other
are equal to each other. For instance, (5 5 5 5 5 5 5 5 3 1 2 5 5) and
5 5 5 5) are examples of such sequence S.
mplexity of the insertion sort, in the worst-case, when sorting a
press the time complexity as a function of k and n. Give also a
orst-case sequence with the format described above. Motivate

ed sequence $S = (1\ 2\ 3\ \ldots\ k\ x\ x\ x\ x\ \ldots\ x)$ with $n > 0$
$> k$ is a constant. Indicate the time complexity of the
S, using the first element as the pivot. Express the
nd n. Motivate your answer.

Part II

Exercise 4 *[10p]*

Consider a set S of n elements, such that $20 < n < 5000$ and each element has a unique key. Describe a data structure to represent set S such that the following operations are supported.

1. To search for an element in S, given its key.
2. To insert and remove an element in S.
3. To list all elements in the set S by reverse order of their insertion. Thus, if x was the last element inserted in S then x is the first to be listed, while if w was the first element inserted in S then w is the last one to be listed.

Indicate also the time complexity for each of the three operations above. Motivate your answer.

Exercise 5 *[10p]*

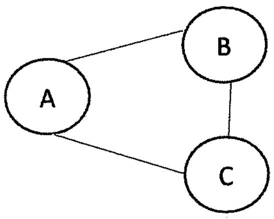
Give an **iterative algorithm** to display the values stored in the nodes of a binary tree **T** in inorder. Assume that each node of the tree stores a value and two pointers, one pointer to the left subtree and another pointer to the right subtree. Note that **T** is not a threaded binary tree.

Present the pseudo-code of your algorithm and indicate its time and space complexity. Motivate your answer.

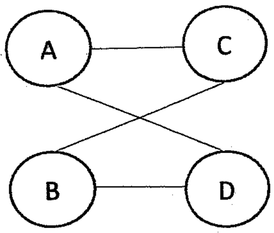
Part III

Exercise 6 *[20p]*

An undirected graph $G = (V, E)$ is a **bipartite graph**, if V can be partitioned in two disjoint subsets, V_1 and V_2, and no edge $e \in E$ has both its vertices in the same subset (i.e. there are no edges between the vertices in the same subset). Some examples are shown below.

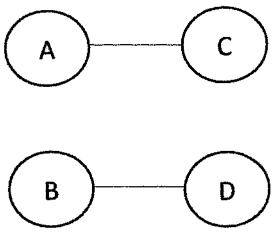


Non-bipartite graph.



Bipartite graph:

$V_1 = \{A, B\}$ and $V_2 = \{C, D\}$.



Bipartite graph:

$V_1 = \{A, B\}$ and $V_2 = \{C, D\}$.

Give an algorithm that for an undirected graph $G = (V, E)$ determines whether G is a bipartite graph. Indicate also the time complexity of your algorithm and motivate your answer.