Linköping University
Department of Science and Technology/ITN
Aida Nordman

Exam: TND004
Data structures

2015-06-25, 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 *[7p]*

Consider two sequences X and Y, with $n \geq 1$ values each, not having any value in common. Furthermore, sequence $X = \langle x_1, x_2, \cdots, x_n \rangle$ is sorted decreasingly, while sequence $Y = \langle y_1, y_2, \cdots, y_n \rangle$ is sorted increasingly. Then, the sequence $X \times Y$ is defined as folows.

$$X \times Y = \langle x_1, y_1, x_2, y_2, \cdots, x_n, y_n \rangle$$

For example, if $X = \langle 8, 5, 4, 3 \rangle$ and $Y = \langle 10, 11, 12, 16 \rangle$ then $X \times Y = \langle 8, 10, 5, 11, 4, 12, 3, 16 \rangle$.

Assume the values of the sequence $X \times Y$ are inserted in the data structures below, following the sequence's order (i.e. x_1 is inserted first, y_1 is inserted second, x_2 is inserted third, and so on).

Indicate the time complexity of the following operations. Use big-Oh notation and motivate briefly and clearly your answers.

a. To create a binary search tree (BST) by inserting the values of the sequence $X \times Y$, starting from an empty BST.

b. To create an increasingly sorted singly linked list by inserting the values of the sequence $X \times Y$, starting from an empty list.

c. To create an increasingly sorted doubly linked list by inserting the values of the sequence $X \times Y$, starting from an empty list.

Exercise 2 *[7p]*

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 sorting algorithm is **stable** if it preserves the initial order of the records with the same key.

Indicate whether the following sorting algorithms are stable and motivate your answers. If some algorithm is not stable then motivate with a concrete example.

a. Heapsort.

b. Insertion sort.

c. Merge sort.

...each, not having any value in common. ...sorted decreasingly, while sequence $Y =$ ..., the sequence $X \times Y$ is defined as follows.

...3) and $Y = (10, 11, 12, 16)$ then $X \times Y = (8,10,5,11,4,12,3,16)$. ...= $(x_1, y_1, x_2, y_2, \cdots, x_n, y_n)$

...order (i.e. x_1 is inserted first, y_1 is inserted second, x_2 is inserted third, and so ...

...reate the time complexity of the following operations. Use big-Oh notation and motivate briefly and clearly your answers.

a. To create a binary search tree (BST) by inserting the values of the sequence $X \times Y$, starting from an empty BST.

b. To create an increasingly sorted singly linked list by inserting the values of the sequence $X \times Y$, starting from an empty list.

c. To create an increasingly sorted doubly linked list by inserting the values of the sequence $X \times Y$, starting from an empty list.

...ercise 2 [7p]

...er 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);
}
```

...cursive mathematical function $f(n)$ that describes what the **mystery** ...calculates.

...mplexity of the **mystery** function? Motivate your answer.

...plexity of the **mystery** function? Motivate your answer.

...serves the initial order of the records with the same key.

...lgorithms are stable and motivate your answers. If ... with a concrete example.

b. Insertion sort.

Part II

Exercise 4 *[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.

Exercise 5 *[10p]*

Consider that there is a file **F** that stores $N > 0$ records and each record has an integer key. Moreover, the file contains records with repeated keys and there are $M > 0$ different possible keys, where $N \gg M$ (i.e. N is much larger than M).

One of your colleagues has asked your help to find the K most frequent keys occurring in **F**, for a given value $0 < K < M$.

You may assume that no two keys have the same number of occurrences in file **F**. Assume also that the memory space required by all records in file **F** is too large to keep all N records in main memory at once (but one can keep M records in main memory).

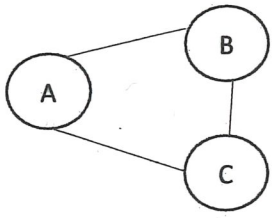
Describe the data structures and algorithms you would use to solve this problem efficiently. Moreover, your solution should be flexible enough such that if the records in another file **S** are added to **F** later on then the data structures can be incrementally updated with the new records in **S**. Thus, one can easily get the K most frequent keys occurring in the updated file (i.e. obtained by adding the records in **S** to file **F**), without the need to process file **F** from the beginning.

Indicate also the time and space complexity of your solution.

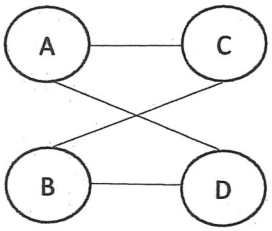
Part III

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.

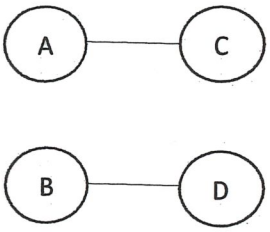


Non-bipartite graph.



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.