# Data Structures and Algorithms

Dragoş Alin Rotaru

Computer Science, University of Bucharest, Romania

# 1 Introduction

These seminar notes contain my overview of the Data Structures and Algorithms course held at University of Bucharest. Because the course is based on heavy theoretic lectures, I tried a more practical approach to present some of the notions by discussing problems which arise natural from the main course.

Most of the problems come from a romanian website specialized on programming contests as well as codeforces or topcoder [1–3]. Of course, there are more interesting problems to tackle, but unfortunately I limit to the course material although sometimes I will talk about some ad-hoc problems.

# 2 Seminar I

Synthesise first 2 courses:

- Basic notions of time and memory complexity.
- Stacks and Queues.

## 2.1 Sketch

What is an algorithm? How can we measure the time complexity of a program? Examples (Choosing every pair of elements and erathosthene sieve). Introduction to stacks and queues. Details about their implementation and a short tutorial in STL. Can also talk about circular queues and double ended queues.

## 2.2 Partial Sums without subtracting

## 2.3 Checking if an expression has brackets in right order

## 2.4 Emulate a queue using 2 stacks

## 2.5 Editor [4]

## 2.6 Alee [5]

## 2.7 Trompeta [6]

Given N digits, find the maximum number with K digits such that the digits preserve the initial order.

### 2.8 Tsunami [7]

Given a matrix NxM with digits less than a value $K(<= 10)$. Find the number of cells with value less than a threshold $T(T <= K)$ such that when you start from 0 you always go up to digits $<= T$.

### 2.9 Add or Multiply

Take a number X. You can add, subtract, multiply it by numbers in a set $S$. Find the shortest number of steps required to reach from $X$ to $Y$.

### 2.10 Take-Out [8]

N blocks, each black or white. You can remove at a time $k$ white blocks and 1 black block such that there is no gap between the removed blocks.

### 2.11 Devices

You are given a row of n devices, each consuming some subset of k¡=8 different resources when turned on, and producing some amount of energy when turned on. For each l from 1 to n you need to find the smallest r such that it's possible to turn on some devices from the segment [l;r] such that no two devices turned on consume the same resource, and that the total energy of the devices turned on is at least z [9].

## 3 Seminar II

– Divide and conquer, merge-sort, estimating complexity

### 3.1 Inv [10]

### 3.2 Stergeri [11]

### 3.3 Muzeu [12]

### 3.4 Binar [13]

## 4 Seminar III

Binary search. Fast exponentiation. K'th element. Time complexity proofs (at least sketches).

### 4.1 Classic Task [14]

### 4.2 Nrtri [15]

### 4.3 Sdo [16]

As a further challenge, consider that you can't store $O(N)$ elements. Can you do dynamic medians?

### 4.4 Loto [17]

## 5 Seminar IV

Heaps. Binary Heaps. Fibonacci Heaps.

Basic heap operations (Insert, Extract-Min, Merge).

Time, space complexity analysis. Heap reconstruction in $O(N)$.

### 5.1 Sdo [16]

Can you find the median using heaps? Can you do this dinamically?

### 5.2 Merge $K$ lists [18]

You have $K$ sorted arrays. Can you merge them quickly?

### 5.3 Sort using heaps

Classic sort algorithm, this time using heaps. Almost sorted list, can you sort it more efficiently than classic?

## 6 Seminar V

Stanford exercises. [19]

Wiki page about AVL. [20]

# 7 Back-up problems

## 7.1 Secv6 [21]

## 7.2 Dynamic GCD [22]

## 7.3 Matrice 2 [23]

## 7.4 Cabane [24]

## 7.5 Minim2 [25]

## References


1. WebSite: Infoarena (2015) https://www.infoarena.ro.
2. WebSite: Topcoder (2015) Last Accessed: October, 2015, https://topcoder.com/tc/.
3. WebSite: Codeforces (2015) Last Accessed: October, 2015, https://codeforces.com.
4. WebSite: Editor https://www.infoarena.ro/problema/editor.
5. WebSite: Alee https://www.infoarena.ro/problema/alee.
6. WebSite: Trompeta https://www.infoarena.ro/problema/trompeta.
7. WebSite: Tsunami https://www.infoarena.ro/problema/tsunami.
8. WebSite: Take out (2015) http://main.edu.pl/en/archive/oi/20/usu.
9. WebSite: Devices (2015) Last Accessed: October, 2015, http://petr-mitrichev.blogspot.com/2015/06/a-week-with-h2.html.
10. WebSite: Inv https://www.infoarena.ro/problema/inv.
11. WebSite: Stergeri https://www.infoarena.ro/problema/stergeri.
12. WebSite: muzeu https://www.infoarena.ro/problema/muzeu.
13. WebSite: Binar https://www.infoarena.ro/problema/binar.
14. WebSite: Classictask https://www.infoarena.ro/problema/classictask.
15. WebSite: nrtri https://www.infoarena.ro/problema/nrtri.
16. WebSite: Statistici de ordine https://www.infoarena.ro/problema/sdo.
17. WebSite: loto https://www.infoarena.ro/problema/loto.
18. WebSite: K sorted lists https://leetcode.com/problems/merge-k-sorted-lists/.
19. WebSite: Stanford exercises (2015) Last Accessed: November, 2015, http://cslibrary.stanford.edu/110/BinaryTrees.html.
20. WebSite: Avl wiki (2015) https://en.wikipedia.org/wiki/AVL_tree.
21. WebSite: Secv6 https://www.infoarena.ro/problema/secv6.
22. WebSite: Gcd http://acm.timus.ru/problem.aspx?space=1&num=1846.
23. WebSite: matrice2 https://www.infoarena.ro/problema/matrice2.
24. WebSite: cabane https://www.infoarena.ro/problema/cabane.
25. WebSite: minim2 https://www.infoarena.ro/problema/minim2.