Data Structures and Algorithms **Spring-Summer 2012/13**

Assignment 3 **Valeria Fionda, Mouna Kacimi,**
Werner Nutt, Simon Razniewski

Heapsort and Matching Pairs

Instructions: Your assignment should represent your own effort. However, you are not expected to work alone. It is fine to discuss the exercises and try to find solutions together, but each student shall write down and submit his/her solutions separately. It is good academic standard to acknowledge collaborators, so if you worked together with other students, please list their names.
You must be prepared to present your solution at the lab. If you are not able to explain your solution, this will be considered as if you had not done your work at all.
You can write up your answers by hand (provided your handwriting is legible) or use a word processing system like Latex or Word. Experience shows that Word is in general difficult to use for this kind of task.
For a programming task, your solution must contain (i) an explanation of your solution to the problem, (ii) the Java code, in a form that we can run it, (iii) instructions how to run it. Also put the source code into your solution document. For all programming tasks, it is not allowed to use any external libraries ("import") or advanced built-in API functions (for example, `String.indexof("a")` or `String.substring(1,5)`), if not stated otherwise.
Please, include name, matriculation number and email address in your submission.

1. Heapsort on Sorted Arrays

Answer the following questions. Justify your answers.

1. What is the running time of heapsort on an array A of length n that is already sorted in increasing order?

2. What is the running time of heapsort on an array A of length n that is already sorted in decreasing order?

3. Is there an initial configuration of an array A of length n that will result in linear running time of heapsort?

(10 Points)

2. Smallest Element of a Max-heap

Suppose you are given an array A of length n that represents a max-heap. What is the cost of an algorithm for finding the smallest element in this array/heap? Justify your answer.

(5 Points)

3. Matching Pairs

Let s be an integer and A an array of integers. Then A has a *matching pair* for s if there are two distinct positions i, j such that $A[i] + A[j] = s$.
The matching pair problem is to check, given s, whether A has a matching pair for s.

1. Develop an algorithm in pseudocode that solves the matching pair problem as fast as possible.
   Hint: An algorithm that you know from the lecture may be useful.

2. What is the asymptotic worst-case complexity of your algorithm?

3. Prove that your algorithm is correct. That is, show that whenever your algorithm returns *Yes*, there are two values in A that add up to b, and that if your algorithm returns *No*, there are no such values.
   Hint: Choose the right loop invariant.

(15 Points)

Submission: Until Wed, 10 April 2013, 11:59 pm, to

```
dsa-submissions AT inf DOT unibz DOT it.
```

If you want to submit a hand-written solution, scan it and send it to the email address above.