Assignment 4

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

Sorting Algorithms and a Mystery

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. Sorting algorithm comparison

In this exercise you are asked to:

- Implement in Java the *Quicksort* algorithm for arrays of integers.
- Implement in Java the *Hybrid Quicksort* algorithm for arrays of integers that. This is a variant of Quicksort that depends on an additional parameter k. The algorithm works as follows:
  - if the size of the portion of the array to be sorted is greater than k it recursively calls the Hybrid Quicksort method with the same k;

  - if the size of the portion of the array to be sorted is less than or equals to k it sorts that portion by using the Insertion Sort algorithm.

- Empirically compare these two sorting algorithms for different values of k (from $k = \{1, 2, ..., 25\}$) and different array sizes. Find out for which values of k Hybrid Quicksort outperforms the standard Quicksort.

Provide: *(i)* a discussion and possible explanation for the obtained running times and *(ii)* analyse for which value of k the standard Quicksort algorithm starts to outperform the hybrid one.

Hint: Consider also to increase the memory size if needed, using the virtual machine option -XX:AggressiveHeap.

(20 Points)

2. A Mystery

Consider the following procedure that takes as input an array and two indices.

```
1:  procedure MYSTERY(A, l, r)
2:      range := r − l + 1
3:      subrange := ⌈2 · range/3⌉
4:      if range = 2 and A[l] > A[r] then
5:          swap A[l] ↔ A[r]
6:      else if range ≥ 3 then
7:          MYSTERY(A, l, l + subrange − 1)
8:          MYSTERY(A, r − (subrange − 1), r)
9:          MYSTERY(A, l, l + subrange − 1)
10:     end if
11: end procedure
```

Note that division in line 3 is division of real numbers and recall that the ceiling function $\lceil x \rceil$ returns the least integer that is greater or equal to x.

1. What effect does the call MYSTERY$(A, 1, A.\text{length})$ have on an array A? Give a proof for your claim.
   Hint: Since MYSTERY is a recursive procedure, your proof should use induction.

2. What is the asymptotic running time of MYSTERY? Provide an argument for your answer.

Submission: Until Sat, 20 April 2013, 11:59 pm, to

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

Submit your work in two files, one PDF document and one .tar or .jar file with your code.