Algorithms and Data Structures
Jacobs University Bremen
Dr. Florian Rabe

Homework 3
given: 2017-02-21

Homework 3

You have to submit your solutions as announced in the lecture.
Unless mentioned otherwise, all problems are due 2017-03-02
There will be no deadline extensions unless mentioned otherwise in the lecture.

Problem 3.1 *Problem Complexity*

Points: 6+6+6

What is the complexity class of the worst-case time complexity $C(n)$ of the following *problems*:

1. Find the smallest element in a list of length n of integers.
2. Find an unknown $x \in \mathbb{N}$ between 0 and n by repeatedly asking yes/no questions about x.
   (Any question is allowed that has a well-defined answer. For example, you may ask "Is x prime?" or "Is $x == 5$?" A solution must determine x with perfect certainty in all cases.)
3. Crack an unknown password if the only prior knowledge you have is that it contains n characters.

In each case, answer the question in two parts:

- Give an algorithm (in pseudo-code or a programming language) that is in the given complexity class. (3 points each)
- Argue informally (but convincingly) why there can be no better algorithm. (3 points each)

Problem 3.2 *Complexity Analysis*

Points: 5

The following is a (highly simplified variant of an) example that came up last year in the instructor's research: Consider the following function in the Scala programming language:

```
def processString(s: String) {
  var rest: String = s
  while (s != "") {
    if (rest.startsWith("foo")) {
      // does not matter, assume this takes O(1)
    } else {
      // does not matter, assume this takes O(1)
    }
    rest = s.substring(1)
  }
}
```

Here `startsWith` and `substring` are methods on strings from the Java library (which Scala can call with only constant-time overhead).

Let $C(n)$ be the run time of this function where n is the length of the input string. What is the complexity class of $C(n)$ and why?

You can try the program yourself on increasingly large input to find out. (If you prefer, you can use Java instead of Scala—the effect is the same.) In the author's case, the surprising effect was noticed by a student trying to process a 100 MB text file, e.g., when $n > 10^8$. The problem is already noticeable for smaller values of n.

Problem 3.3 *Polynomial Algorithms*

Points: ∞

Consider the following problem: Given a natural number x that uses n bits, find a non-trivial factor of x (or say that n is prime).

Here "non-trivial factor" means a number $p|x$ such that $1 < p < x$.

Give an algorithm (pseudo-code or programming language) with polynomial run-time complexity $C(n)$ or show that no such algorithm exists.