Exercises week 1

Complexity 2011-2012

A. Silva, H. Barendregt, B. Westerbaan & B. Westerbaan

These are the exercises that will be discussed in week 1. The exercise marked with (†) is a hard exercise: every week we will include one or two exercises at a more advanced level. Exercises marked with a **(*)** can be handed in, we will correct them and give them back to you the week after. This week the answers should be handed in till **Wednesday February 1, 17h00**.

Handing in your answers: There are two options:

1. E-mail: Send your solutions by e-mail to `alexandra@cs.ru.nl` with subject '*Complexity - assignment*'. This e-mail should only contain a single PDF document as attachment. Before sending an e-mail make sure:
   - the file is a PDF document
   - your name is part of the filename (for example MyName-week1.pdf)
   - your name and student number are included in the document (since they will be printed).
2. Post box: Put your solutions in the post box of Alexandra. The postbox is located in the second floor, Wing 5, of the Huygens building, where computer science is located. Wing 5 is off the main corridor, where the Intelligent Systems group resides. Silva's personal mailbox is at the end of wing 5 (just next to Herman Geuvers' office). Before putting your solutions in the post box make sure:
   - your name and student number are written clearly on the document.

The rooms for the werkcollege are EL 2.55 en EL 2.56, laagbouw van het Erasmusgebouw.

Exercise 1. (*) Find f, g such that $f \not\lesssim g$ and $g \not\lesssim f$.

Exercise 2. Show that

(i) $2^{n+1} \in O(2^n)$.

(ii) $2^{2n} \notin O(2^n)$.

(iii) $n^{10} \in O(2^n)$. (**Note:** this is also a harder exercise, we shall discuss it in class.)

Exercise 3. Let f and g be functions that take non-negative values, and suppose $f \in O(g)$. Show that $g \in \Omega(f)$.

Exercise 4. (*) Suppose you have algorithms with the six running times listed below. Suppose you have a computer that can perform 10^{10} operations per second, and you need to compute a result in at most one for of computation. For each of the algorithm, what is the largest input size n for which you would be able to get the result in one hour?

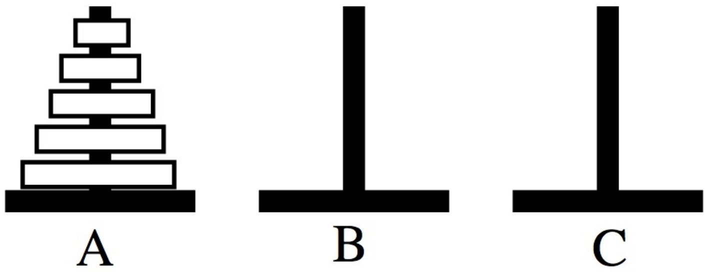
(i) n^2. (ii) n^3. (iii) $100n^2$. (iv) $n \log n$. (v) 2^n. (vi) 2^{2^n}.

Exercise 5.

(i) Let Σ be a finite alphabet whose elements are ordered. For $v, w \in \Sigma^*$ (words over Σ), there is an algorithm to determine whether $v <^* w$, where $<^*$ denotes the lexicographic order (dictionary like ordering). Determine the complexity of the algorithm.

(ii) Repeat for $v, w \in \mathbb{N}^*$, this time the complexity in terms of the length of the list and it's largest element.

Exercise 6. (†)



The Towers of Hanoi is an ancient puzzle, originating with Hindu priests in the great temple of Benares. Suppose we are given three towers, or more humbly, three rods, and a number of disks of different sizes which can slide onto any rod. The puzzle starts with the disks in a neat stack in ascending order of size on one rod, the smallest at the top, thus making a conical shape (see image above). The objective of the puzzle is to move the entire stack to another rod (say from A to B), obeying the following rules:

- Only one disk may be moved at a time.
- Each move consists of taking the upper disk from one of the rods and sliding it onto another rod, on top of the other disks that may already be present on that rod.
- No disk may be placed on top of a smaller disk.

If there are three discs in A the puzzle can be solved as follows: move A to B ; move A to C; move B to C; move A to B; move C to A; move C to B; move A to B. Hence, in 7 steps.

The inventors of the puzzle stated that the world will end if you solve the above puzzle with 64 rings!

Write a procedure/algorithm that solves the above puzzle and give an estimate on the number of moves one needs to solve the Hanoi puzzle.