Exercise 0, Discrete Mathematics for Bioinformatics

Sascha Meiers, Martin Seeger

Winter term 2011/2012

0.1 Background Reading I

The six (not seven) statements:

1. *Conceptualizing, not programming.* Agree.
2. *Fundamental, not rote skill.* Disagree.
3. *A way that humans, not computers, think.* Disagree.
4. *Complements and combines mathematical and engineering thinking.* Agree.
5. *Ideas, not artifacts.* Agree.
6. *For everyone, everywhere.* Disagree.

Statement of strongest agreement: *Ideas, not artifacts.*
Jeanette Wing states that computational concepts will be "present everywhere and touch our lives all the time", besides also the "software and hardware artifacts" that represent them. We would like to argue that this time has already come by means of two prominent examples:

- Social networks are an area in which graph theory has invaded human thinking (and probably vice versa). Thanks to the ubiquity of social networks, and their mobile availability, participants have stopped thinking about the hardware or software on which Facebook or other social applications are running. However, they do (usually unknowingly) think about graph theoretical concepts such as connectivity, distances etc. which was certainly less common before the rise of social networks.
- Similarly, navigation and geolocation systems help manage our daily lives and make (some of) us think about a whole other range of problems, e.g. how to calculate shortest paths.

One could add further examples, such as optimization and search problems which have recently arrived in a large part of the broad population and have begun to shape their thinking.

Statement of strongest disagreement: *For everyone, everywhere.*
Having argued in the previous paragraph that computational concepts are ubiquitous and influence our lives already now, we would now claim that this does probably not happen in the way the author envisages when she says it might become "so integral to human endeavors" that "it disappears as an explicit philosophy". It is wishful thinking to believe that more than a minority of the population would significantly change their attitude towards computational thinking which she likens to mathematical and engineering thinking in one of the earlier statements.

Realistically, the majority of the population will not change their style of thinking and their attitude towards mathematics or computer science by more exposure to social networks, for example, or to other manifestations of computing techniques. For this to happen, profound and significant changes would have to occur in early education and even then "computational thinking for everyone" will remain an illusion. People will stay the way they are, pretty much, no matter how much computation is happening around them.

0.2 Background Reading II

Let an arbitrary sequence $s = \{x_1, \dots x_n\}$ be given. Using Merge Sort, this sequence can be sorted in $\Theta(n \lg n)$.

Let a number y be given. The task is to determine whether s contains multiple occurences of y. This is done with the following pseudocode:

```
z_1, ..., z_n = merge_sort(x_1, ..., x_n)
for i = 2 to n:
    if y==z_i and y==z_(i-1): print "multiple occurence:", y
```

This codes has total runtime $\Theta(n \lg n) + \Theta(n) = \Theta(n \lg n)$.

If the task is, however, to determine whether any of the x_i occurs in s multiple times, then the following pseudocode can be applied:

```
z_1, ..., z_n = merge_sort(x_1, ..., x_n)
for i = 2 to n:
    if z_i==z_(i-1): print "multiple occurence:", z_i
```

Again, the runtime is $\Theta(n \lg n)$.

0.3 Background Reading III

Example of a Monte Carlo algorithm: numerical integration, in particular in high dimensions.
Example of a Las Vegas algorithm: Randomized Quick Sort.