Issue Date: 20.04.2015
Version: 2015-04-10 16:39

1st Hands-on sheet for Advanced Algorithmics, SS 15

Hand In: *in lecture, exercise sessions, hand-in box in stairwell 48-6 or via email.*

What we do here

In addition to the "classic" exercise problems, you may want to study the material from a more applied perspective. Particular concerns are comparing the "practical" performance of algorithms (which might use very different paradigms), and validating theoretical models.

The hands-on scenarios are to provide a common guide towards this goal. We will consider two selected "real-world" problems and try to apply the techniques learnt in lecture to them. We expect that you might need to do independent literature research to fully solve the problems.

Note that this track of exercise is completely optional, not required in exams (though hopefully helpful) and not covered by class credit. Proceed at your own risk and leisure, and – most importantly – have fun!

In order to encourage and facilitate collaboration, we have set up a shared code repository:

`https://github.com/reitzig/advalg15`

We invite you to share your work there.

Skills

Hand-on problems require a skill set slightly different from the one needed for the exercises. While prior exposure is certainly advantageous, we think you can train them on the go.

- Knowledge of the *lecture material.*

- *Research* ability. We will assume that you will not try to develop algorithms yourself but look up prior work in literature[1].
- *Programming* and *program testing.* If we are to talk about the performance of algorithms, we need to have correct implementations first. Choose a language you are comfortable with.
- Setup of *experiments* and appropriate *analysis* of their results. You might want to check out "A Guide to Experimental Algorithmics" by C. C. McGeoch for many fundamental skills and advice.

Scenario A: Zombies in Kaiserslautern

Zombies[2] are loose! Lucky for us, the U. S. military is close by and has an interest in pacifying Kaiserslautern. They have limited resources, though, and therefore need to develop an efficient deployment plan.

We have the following restrictions:

(I) Patrols are risky, so we favour entrenched positions with all-round vision. We will assume that we can station units on roads and crossroads.

(II) We can allow neither the undead nor plunderers any reprieve. Therefore, we want to have every stretch of road and crossroads covered.

(III) We want to use as few units as possible.

Scenario B: Clearing snow in Kaiserslautern

As everybody who has spent a winter season in Kaiserslautern knows, our snow-plowing service has only one vehicle at its disposal. Since this is not nearly enough to clear *all* snow away, decisive measures have to be taken. From now on, we will no longer try to ensure that every street is cleared, but only that we can get everywhere *somehow.*

We need a route for the snow-plower that ensures the following.

(I) After the tour is complete, there has to be a snow-clear route from every crossroad to every other.

(II) The snow-plower takes the fastest route possible.

[1] That is not to say that trying won't be helpul and appreciated, but it definitely costs time. We want to focus on working *with* algorithms in this track, so do not beat yourself up if you don't have the ideas yourself. They are going to be tough.

[2] If you are uncomfortable with the idea of undead roaming campus, assume soccer fans instead.

Step 1: Modelling

How can we model either scenario formally? In particular, try to answer the following questions:

1. Which formal model is suitable for the underlying data?
2. How can we express the goal in terms of this model? Does our problem fit any well-studied abstract problems?
3. Which simplifying assumptions do we have to make? How do they influence the applicability of any result we obtain?
4. How hard is the problem, algorithmically?

Step 2: Data

Obtain suitable example data from the course repository. It is in a format not immediately suitable for our purposes and may contain irrelevant detail and/or "noise"; that will have to be remedied.

- Transform the data into a suitable format, be it in some data structure or file.
- Strip the data down to their essence.
- Visualise the data so we get a feeling of what we are looking at.
- Do the data have striking features? Both visual and computational analysis may be of use.

The data are three street graphs obtained from the OpenStreetMap dump[3]. They contain all of Kaiserslautern, Kaiserslautern city center and university campus plus residential area, respectively. They are provided in the `OSM` flavor of XML[4].

[3] See openstreetmap.org and download.bbbike.org/osm.

[4] See wiki.openstreetmap.org/wiki/OSM_XML for detail.