

Remarks on Supervisions for 1A Lent Algorithms

Hayk Saribekyan (hs586)

January 19, 2019

1 General

- The supervision schedule is below:

<https://goo.gl/mbzeUh>

You may find that the Agenda view is better. Let me know if changes need to be made at any point.

Note that the locations of most of the supervisions is set.

- For each supervision, please submit your work scanned in a pdf format using the following link:

<https://www.dropbox.com/request/mQHL09b26Cyi15Tytv8U>

The name of the file should contain your CRSid and the number of the supervision. Please submit your solutions by 17:00 the day before the supervision.

This may be a good opportunity to learn LaTeX, which is essential in postgraduate studies and beyond. Learning may take some time, but in the long-term will help you a lot.

- Indicate in the margin parts of your solutions that you would like to discuss during the supervisions. Also, regardless of whether you solved a problem or no, feel free to indicate that you would like to discuss it. A simple question mark in the margin is enough. For LaTeX users the package `todonotes` may come handy.

The point of this is not mainly for me to prepare and allocate time properly. It is for you to remember during the supervision what questions you had. Time is limited during the supervision and during quick changes of topics, it is easy to forget things.

- Spend few minutes before the supervision looking at your work. This way you will not spend the first 10 minutes of the supervision trying to remember what the problems were about. Also your question marks from the previous bullet point will help. It is tempting to skip this step, but don't :)

2 Algorithms Specific

- Giving concise and correct explanations of your algorithms is essential. For now it is to get higher exam marks, later on you will need to do it for grander purposes (job interview...).

Explaining algorithms in written form is not easy because there are two things involved (1) the steps of the algorithms and (2) what happens during these steps e.g. how the state of the memory changes and what for. During the supervisions (2) is mostly done orally, therefore it is easier. However, you will not have this luxury during the exam, so I urge you to write details even if you find them trivial sometimes.

Pseudocodes and bullet points are your friend. Look at CLRS and your lecture notes for examples.

- I encourage you to code some of the algorithms you have learnt in this course, despite the fact that this is not a requirement for the course. It will help your understanding of the concepts. The programming language is not important.

If you want to go a step further, you can start solving problems from online judges. Your solutions will get tested immediately and you will know whether your algorithm (and implementation) are correct.

After writing a few simple programs, a good places to move on to are CodeForces, Timus and SPOJ. The latter two have an amazing selection of problems of any difficulty and are great for practice. In SPOJ you can filter problems by the type (e.g. sorting algorithms, dynamic programming, etc). CodeForces is great for online contests, which are great fun.

- If you would like to discuss a problem outside of the example sheet, e.g. from an online judge, let me know (ideally beforehand).
- Often I will suggest supplementary problems for you to work on. These can be from online judges and would require you to implement the algorithm in some programming language. I would be happy to discuss these problems, but it is up to you whether you want to work on them or no. You may not have the time now, but you could sharpen your algorithms intuition on them during the breaks.