University of Glasgow | School of Computing Science

Urban Data Timeline

Yordan Yordanov

School of Computing Science
Sir Alwyn Williams Building
University of Glasgow
G12 8QQ

Level 4 Project — March 28, 2016

Abstract

We show how to produce a level 4 project report using latex and pdflatex using the style file l4proj.cls

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic format. **Please note that you are under no obligation to sign this declaration, but doing so would help future students.**

Name: _____ Signature: _____

Contents

Chapter 1

Introduction

The first page, abstract and table of contents are numbered using Roman numerals. From now on pages are numbered using Arabic numerals. Therefore, immediately after the first call to \chapter we need the call \pagenumbering{arabic} and this should be called once only in the document.

The first Chapter should then be on page 1. You are allowed 50 pages for a 30 credit project and 35 pages for a 20 credit report. This includes everything up to but excluding the appendices and bibliograph, i.e. this is a limit on the body of the report.

You are not allowed to alter text size (it is currently 11pt) neither are you allowed to alter the margins.

Note that in this example, and some of the others, you need to execute the following commands the first time you process the files. Multiple calls to pdflatex are required to resolve references to labels and citations. The file bib.bib is the bibliography file.

```
> pdflatex example0
> bibtex example0
> pdflatex example0
> pdflatex example0
```

1.1 First Section in Chapter

The quick brown fox jumped over the lazy dog [1].

1.1.1 A subsection

The quick brown fox [3] jumped over the lazy dog. The quick brown fox jumped over the lazy dog.

Chapter 2

Related Work

There are applications that have been developed to accomplish similar if not the same goals, but most of them are concentrating on a single data stream. This chapter is presenting a short description of related products, along with key differences and correlations in comparison with the goals of this project.

2.1 The Fox Jumps Over

The quick brown fox jumped over Uroborus (Figure 2.1).

The quick brown fox jumped over [2] the lazy dog.

2.2 The Lazy Dog

The quick brown fox [4] jumped over the lazy dog.

Google Maps Timeline [6]

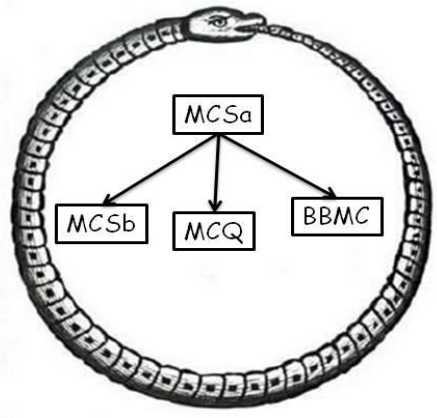

Figure 2.1: An alternative hierarchy of the algorithms.

Google stores a history of where anybody that uses its location services goes. All this data is collected by your device sensors and the navigation you use and then visualized with the Google Maps Timeline feature. It is advertised to be an easy way to view and remember places somebody has been on a given day at a given time. Without any input, the timeline shows predictions of when you have arrived or left a place. One of the key features is highlighting when you have visited the most places and how you had travelled. The application does not offer sharing as Google wants your information to remain private. As far as human interaction goes, you are allowed to correct, confirm or delete a place where Google thinks you had been but even after deleting, it can still be seen that you had passed by that area. In 2009, the company released a similar feature called Google Latitude that offered sharing of location history but the project was closed down [5].

This product has the same initial goal to visualize timely data originating from different streams like navigation history, pictures, travel and walking routes. However what it has in common is it uses a vertical timeline to display the events, it allows searching based on a specific day, month, year and has a way to show (via bar chart) how active you have been every day in the past few weeks.

Social Media Timeline

All social media websites in the likes of Twitter and Facebook use some sort of a timeline to visualize their user's personal information. However they differ on the way they organize their posts. Facebook describes a timeline to be a place on your profile where you can see your own posts, your friends' activities and stories you're tagged in, sorted by the date and time they were posted. On the other hand, Twitter displays a stream of tweets from accounts that you have chosen to follow. Making use of machine learning algorithms, posts that you are likely to care about more are displayed first.

Despite of the fact that social media applications has as a main goal the delivery of a secure and reliable tool for communication, they also provide their users with the ability to visualize their personal data streams like photos, events, group activities and accomplishments. Both Facebook and Twitter use a vertical timeline and allow for searching based on keywords. One of the key features is infinite scrolling that make the illusion of one endless stream of events by making use of the million users, posting every day.

Chapter 3

Solution

The quick brown fox jumped over the lazy dog.

3.1 Choice of technology

3.2 Components communication

Chapter 4

Components

4.1 Model

4.2 View

4.3 Controller

4.4 Challenges

Chapter 5

Evaluation

5.1 Product evaluation

5.1.1 Initial phase

5.1.2 Final phase

5.2 Testing

Chapter 6

Future work

Appendices

Appendix A

Running the Programs

An example of running from the command line is as follows:

```
> java MaxClique BBMC1 brock200_1.clq 14400
```

This will apply $BBMC$ with $style = 1$ to the first brock200 DIMACS instance allowing 14400 seconds of cpu time.

Appendix B

Generating Random Graphs

We generate Erdós-Rënyi random graphs $G(n, p)$ where n is the number of vertices and each edge is included in the graph with probability p independent from every other edge. It produces a random graph in DIMACS format with vertices numbered 1 to n inclusive. It can be run from the command line as follows to produce a clq file

```
> java RandomGraph 100 0.9 > 100-90-00.clq
```

Bibliography

[1] DIMACS clique benchmark instances. ftp://dimacs.rutgers.edu/pub/challenge/graph/benchmarks/clique.

[2] Peter Cheeseman, Bob Kanefsky, and William M. Taylor. Where the really hard problems are. In *Proceedings IJCAI'91*, pages 331–337, 1991.

[3] Torsten Fahle. Simple and Fast: Improving a Branch-and-Bound Algorithm for Maximum Clique. In *Proceedings ESA 2002, LNCS 2461*, pages 485–498, 2002.

[4] Brian Hayes. Can't get no satisfaction. *American Scientist*, 85:108–112, 1997.

[5] Barry Schwartz. Google maps timeline: User-friendly location history. http://searchengineland.com/google-maps-timeline-user-friendly-location-history-225783, July 2015.

[6] Google Maps Timeline. https://www.google.co.uk/maps/timeline. Accessed: 2016-03-02.