Report assignment  
of   
Combinatorics and Graphs

04/2020

# Overview

One of the applications of graph theory is social network modeling. In this report assignment, you need to self-study and make a report as well as a program about the social network analysis problems and their existing solutions. The document needs to be in the Faculty’s form and follow the given outlines in part 2. The program is a demonstration of at least 5 algorithms or formulas in your document. It can be done in any programming language.

This report assignment is considered as your final exam. It is a personal assignment and plagiarism is prohibited. Any case of plagiarism will get 0 for this report assignment.

The deadline is at the end of May 15th, 2020.

# Report document

You need to write a report document in the Faculty’s form and follow the below outlines. Using the given instructions and keywords to search for related references. However, you need to rewrite all of that knowledge in your own words (do not just copy and paste).

Besides, you need to give 3 examples, other than ones in your references, for each algorithm or formula or difficult concept. These examples should be given in details, i.e. step by step calculation. Any other related knowledge could also be included in your report if necessary.

English is required for high-quality classes.

## Chapter 1: Social networks and Social network analysis problems

In this chapter, you need to introduce the concept of the social network, how it is important, and how to model it using a graph.

You also need to introduce some problems in social network analysis. However, your report only focuses on 2 problems in chapters 2, and 3.

## Chapter 2: Social network measurements

In this chapter, you need to show a way to describe a social network using a graph. Your knowledge about graph theory can be used here.

Some concepts and measurements of this kind of graph will be explained, including:

* density, degree centrality, closeness centrality, betweenness centrality, clustering centrality;
* key players and how to detect key players;
* signed graph, its problems, and applications.

## Chapter 3: Community detection

In this chapter, you need to describe the problem of community detection, as well as some related concepts and algorithms, including:

* Modularity and cut
* Grivan Newman’s algorithm
* Node similarity based algorithm
* Label Propagation Community Detection (LPA).

## Chapter 4: Demo

This chapter is used to explain your program, which algorithms/formulas you implement, how you implement them, how to use your program… Do not just copy your code here.

## Chapter 5: Conclusion and Discussions

Summerize what you have reported so far. You may give your ideas about any social network analysis problems, solutions, and applications.

## References

All of your references (books, papers, websites etc.) should be included here. Be careful of the format.

## Self-evaluation

You need to evaluate your self according to the below form. This self-evaluation is used for reference only, your final result may not equal this result.

Please read the requirements and their evaluation levels carefully and plan your implementation targets.

Please evaluate yourself in the "Self-evaluation" column. If the score of a requirement is low, students should state the reason in the column "Reason(s)", for example, not understanding the goal, not having enough time, not being able to do it, and not being interested…

**SELF-EVALUATION FORM**

| **Requirements** | **Score /10** | **Level 1** | **Level 2** | **Level 3** | **Self-evaluation** | **Reason(s)** |
| --- | --- | --- | --- | --- | --- | --- |
| **0 score** | **1/2 score** | **Full score** |  |  |
| **1/ Report** | **8.0** |  |  |  |  |  |
| In right format | 1.0 | Wrong format and outlines | Some errors | In right format and outlines, no error |  |  |
| Chapter 1 | 1.0 | Not enough content, bad written, no example | Full contents, not very well written, not enough examples | Full contents, well written, with examples |  |  |
| Chapter 2 | 2.0 | Not enough content, bad written, no example | Full contents, not very well written, not enough examples | Full contents, well written, with examples |  |  |
| Chapter 3 | 2.0 | Not enough content, bad written, no example | Full contents, not very well written, not enough examples | Full contents, well written, with examples |  |  |
| Chapter 4 | 1.0 | Not enough content, bad written, no example | Full contents, not very well written, not enough examples | Full contents, well written, with examples |  |  |
| Chapter 5 | 0.5 | Not enough content, bad written | Full contents, not very well written | Full contents, well written |  |  |
| References | 0.5 | No reference | Wrong format, < 3 references | Right format, ≥ 3 references |  |  |
| **2/ Demo** | **2.0** |  |  |  |  |  |
| Contents | 1.0 | Demo ≤ 1 fomula /algorithm | Demo up to 3 fomulas or algorithms | Demo ≥ 5 fomulas or algorithms |  |  |
| Program | 1.0 | Cannot be compiled | Runtime error for 1 fomula or algorithm | Can be run correctly with no error |  |  |
| **Total** | 10.0 | Result: | | |  |  |

# Demo

You need to write a program to demonstrate at least 5 algorithms or formulas in your document. Any programming language can be used. You can use any supporting library. GUI is not required.

# Submission

Make a folder named with your StudentID. Make 3 other subfolders inside, Report and Demo. Your document, in .doc/docx and .pdf, is put in Report folder. Your source code, resource, and release are put in your Demo folder.

* <StudentID>
  + Demo
    - SourceCode
    - Resouce
    - Release
  + Report

Now compress everything into a single file named with your StudentID, i.e.

<StudentID>.rar/zip

before submitting.

-----

Good luck.