Project 1-1: GRAPH COLOURING

04-10-2018 / 25-01-2019

Courses

Introduction to Data Science and Knowledge Engineering (KEN1110)
Introduction to Computer Science 1 (KEN1120)
Discrete Mathematics (KEN1130)
Knowledge Presentation and Cognitive Psychology (KEN1210)
Introduction to Computer Science 2 (KEN1220)
Linear Algebra (KEN1410)

Graph Colouring

PROJECT 1-1

Summary. A *graph* G is essentially a collection of points connected by lines. The *chromatic number* of G is simply the minimum number of colours you need to colour the points, such that no two points that are connected by a line have the same colour.

- In the first phase of the project (which lasts <u>a single week</u>) a computer program is to be implemented in Java which reads in a small graph from a text file and then computes the chromatic number, or - if this is deemed too computationally intensive – generates lower and upper bounds on the chromatic number. The examiners will provide a suite of test graphs.

- In the second part of the project a game will be implemented in which a human player is confronted with a random graph and is asked to colour the graph with as few colours as possible. The game has different playing modes depending on the way the performance of the player is measured. In all modes the human player should be able to ask for computer-assisted support.

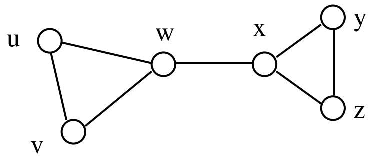
- In the third phase of the project the work of the first phase will be continued. However, this time the graphs provided by the examiner will be potentially much larger and/or have hidden structure that makes it much easier to compute the chromatic number than at first glance. The code developed in this third phase will be entered into a competition in which all the groups compete against each other on a hidden set of test graphs developed by the examiners. In this phase students can optionally consider using a simple machine learning framework to learn which algorithm to use on which graphs.

Contents

1. Project description

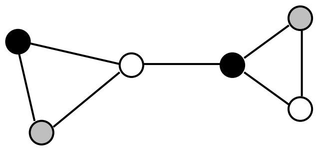
A graph G=(V,E) is simply a set of points V (known as *vertices*) connected by lines E (the *edges*). Each edge connects exactly two vertices. In an *undirected* graph there is no orientation on the edges so E is often modelled as a set of size-two subsets of V. In this project we will deal exclusively with undirected graphs and for brevity will henceforth omit the prefix "undirected". We say that two vertices u and v are *adjacent* if E contains an edge {u,v}. Note: a vertex is not allowed to be connected to itself, and it is not permitted to have more than one edge between a given pair of vertices.

Here is an example of a graph on six vertices V={u,v,w,x,y,z} and seven edges E={{u,v},{u,w},{v,w},{w,x},{x,y},{y,z},{x,z}}. This graph is *connected* because it consists of one piece; graphs can also be *disconnected*, consisting of multiple *connected components*.



Graphs are one of the cornerstones of mathematics and computer science. Think, for example, of road/train networks, the wiring of the brain, social interaction networks...these are all graphs! Given their prominent role in applied computing it's important to be able to develop software/algorithms that can query, explore and manipulate graphs efficiently.

Here we explore the topic *graph colouring*. A colouring of a graph G is an assignment of colours to its vertices such that no two adjacent vertices have the same colour. We are particularly interested in colourings that use as few different colours as possible. The *chromatic number* of G, often denoted $\chi(G)$, is defined as the minimum number of colours required to colour G. The chromatic number of the graph shown earlier is 3, because 3 colours are sufficient (as shown below), but 2 are not.



There are good reasons for wanting to know the chromatic number of a graph. For example: automated timetabling and scheduling. If we let V represent a set of university courses, and E represent "these two courses cannot take place at the same time", then the chromatic number represents the minimum number of time slots required to schedule all the courses. Computing this minimum thus helps to make optimized use of limited resources (time, space, money). As you will discover, computing the chromatic number exactly is surprisingly challenging. It is therefore a good case study in developing code, data structures and algorithms to tackle difficult computational problems.

Phase 1:

In Block 1.1 we want you to write a Java code which reads a small graph G in from a text file and then, ideally, computes the chromatic number of G *exactly*. If you encounter a graph that is too large or complex for your code to cope with, then you should aim to output both lower and upper bounds on the chromatic number that are as tight as possible. (Lower bounds are important because they give you some idea of how close your upper bounds are to the chromatic number).

To be more specific: to pass this phase, you need to develop at least one exact algorithm (i.e. an algorithm that computes the chromatic number exactly) and at least one upper bounding algorithm (i.e. an algorithm that is not necessarily guaranteed to use a minimum number of colours, but which nevertheless uses "few" colours). To achieve a higher grade you also need to code a lower bounding algorithm (i.e. an algorithm that produces some non-trivial lower bound on the chromatic number).

As a general rule, the greater the variety, sophistication, effectiveness, and accuracy of the algorithms you develop, the higher your score.

The code should be your own: do not use third party software and do not cut and paste code found on the internet.

The examiners will provide a test suite of graphs (of increasing difficulty) upon which you can test your code. At this stage you have not been learning Java for long, so the examiners will provide some ready-made Java code that reads the graph from the file into memory for you...but the rest you have to do yourself! Both the test suite and the file-handling Java code are available at:
http://skelk.sdf-eu.org/graphcolouring2018

In the demonstration you will be asked to present your approach(es), your results for the test suite and to explain briefly why your results are mathematically correct. For example, if you claim that the chromatic number of a graph cannot be 3 or less, then you should be able to explain why! Your demonstration should include a short PowerPoint presentation in which you summarize the algorithms you have coded and your results. You might also be asked to execute your code and to show the source code.

Given that this phase of the project is only one week long, we do not expect hyper-optimized code at this stage, but any insights and optimizations you can make at this stage will help you in phases 2 and (in particular) phase 3.

Phase 2:
In Block 1.2 we want you to build a computer application with a user-friendly interface to play a simple game based on computation of chromatic number. This phase links to skills you will learn in (in particular) Introduction to Computer Science 2 (ICS2).

The core of the game is as follows. There is a common core to ensure comparability between project groups. Beyond the core you are free to customize and extend the game as you see fit.

- It is a game for one, human player.

- The computer generates a random graph which is shown on the screen. (The user should be able to specify some basic characteristics of the graph beforehand e.g. number of vertices, number of edges etc.) Alternatively, the user should be able to read in a graph from a file.

- The player is challenged to colour the graph with as few colours as possible. He/she does this by (for example) clicking on vertices and indicating which colour they should have. The computer should warn the user if their current colouring is not valid i.e. if there exist two adjacent vertices that have the same colour.

- There are three different game modes possible. All three should be implemented.

    - *To the bitter end.* The player simply has to colour the graph as quickly as possible. The computer does not allow the player to finish until the minimum number of colours has been reached. (This, of course, means that your code should be able to internally compute the chromatic number…)
    - *Best upper bound in a fixed time frame.* The player is given a fixed amount of time and they have to find a colouring with as few colours as possible in the given time. (Here it is not necessary that the user finds the minimum number of colours).
    - *Random order.* Here the computer generates a random ordering of the vertices and the player has to pick the colours of the vertices in exactly that order. Once the colour of a vertex has been chosen, it cannot be changed again. (Note that the player can never get stuck, because they are always allowed to introduce a new colour.) The goal for the player is to use as few colours as possible.

- In each game mode there must be a 'hint' button which, when pressed, gives the user some computer-assisted help. For example, the computer might propose which vertex should be coloured next, or indicate a subset of colours that should be considered (or avoided) for a given vertex. More sophisticated hint functions will contribute to a higher grade.

Phase 3:

The third phase is conceptually similar to the first phase. However, this time the test suite of graphs provided by the examiner (at the beginning of phase 3) will be larger and more computationally challenging, so you will need to think more deeply about how to optimize your algorithms. To make life more interesting, some of the graphs will have a special structure which makes computation of the chromatic number easier than you would expect - assuming you know how to detect, and then exploit this special structure! For example, if a graph is a *tree* (i.e. has no cycles) then its chromatic number will be (at most) 2, no matter how many vertices or edges it has. There are lots of other special cases. You should explore ways to both enhance the efficiency of your algorithms and to detect these "nice" special cases. If you wish you can re-use the GUI from phase 2, although this is not essential.

If you wish, you can adopt a simple machine learning approach to learn which of your algorithms work best on which graphs. (This is called the *algorithm selection* problem for graph colouring). However, this is not essential – and there are no guarantees it will work well! It might be a better approach to spend your time aggressively optimizing a smaller number of algorithms, and to hard-code rules for deciding which algorithm to use on which graph. The choice is yours.

At the end of the third phase the implementations of the different groups will be compared on a <u>hidden</u> third dataset that the examiners will *not* disclose. To make this possible we will ask you to provide a version of your code without a GUI that reads input and writes output in a very specific way; you *must* adhere to these conventions. (This convention will be announced in due course). There will be a first, second and third prize – and eternal glory - for the groups whose implementations perform best on this hidden dataset.

May the odds be ever in your favour!

Project examiners: P. Bonizzi, S. Kelk, C. Seiler, E. Smirnov

2 . Project phases

The project assignments in block 1.1 and 1.2 are prepatory for project 1-1 which takes place in the third block. The students work during week 6 full-time on the project. This week starts with a project opening on Monday. On Tuesday and Thursday there is a project meeting which is also attended by the tutor (project coordinator).

October 14 the students give a demonstration of their product. The audience consists of the examiners and the tutor. After this demonstration, the group prepares a short note containing the remarks of the lecturers and the actions which result from these remarks.

The demonstrations take place in room BOU8-10 according to the schedule below:

Room 2.002 BOU8-10	**Room 2.015 BOU8-10**
11:00 - 11:15 Group1	11:00 - 11:15 Group15
11:15 - 10130 Group2	11:15 - 11:30 Group16
11:30 - 11:45 Group3	11:30 - 11:45 Group17
11:45 - 12:00 Group4	11:45 - 12:00 Group18
12:00 - 12:15 Group5	12:00 - 12:15 Group19
12:15 - 12:30 Group6	12:15 - 12:30 Group20
13:00 - 13:15 Group7	13:00 - 13:15 Group21
13:15 - 13:30 Group8	13:15 - 13:30 Group22
13:30 - 13:45 Group9	13:30 - 13:45 Group23
13:45 - 14:00 Group10	13:45 - 14:00 Group24
14:30 - 14:45 Group11	14:30 - 14:45 Group25
14:45 - 15:00 Group12	14:45 - 15:00 Group26
15:00 - 15:15 Group13	15:00 - 15:15 Group27
15:15 - 15:30 Group14	15:15 - 15:30 Group28

The groups are requested to be present 10 minutes in advance. The examiners and tutor will be present at these demonstrations.

In block 1.2 students will work minimally 4 hours a week during 7 weeks on the project. Each week there is one project meeting with the tutor. At the end of block 1.2 a short presentation is given for the examiners and tutor.

Good project work in block 1.1 and 1.2 will prove to result in a head start for project 1-1 in block 1.3.

During the first 3 weeks of block 1.3 the students work full-time on the project. At the end of this period each group hands in five copies of its project report to the project coordinator. At the same time, the source code is handed in on CD / DVD / memory stick. Finally, the product, the report as well as the sheets of the presentation have to be placed on \\unimaas.nl\education\FSE_DKE . Finally, a presentation of 20 minutes is given by each group. The audience consists of fellow students, the examiners, and the tutor.

3. Project presentation

On **January 24** each group gives a presentation of at most 20 minutes. The audience consists of lecturers and fellow students. A time schedule as well as location will be announced

4. Project assessment

The project period 1 and 2 will be concluded with a demonstration or a presentation of the results thus far. For the students this demonstration / presentation means feedback on their interpretation of the project assignment, their conclusions and their planning for the next period. These demonstration / presentation of period 1 and 2 enable the lectures to judge if more information and tutoring is needed. At the end of period 3 the final presentation will take place in which the students communicate what their results and conclusions on the project assignment are. The demonstration / presentation of the first and second phase and the final phase have different weights in the total assessment of the block according to 1 : 1 : 4.

The final project result will be a group result unless a student hasn't shown enough attendance and participation. When a student has not participated enough, the examiner can decide to give him or her an additional assignment. The examiner can, in individual cases, decide to deviate from the group result in a positive or negative direction. Attendance to project meetings and project skills is mandatory, unmotivated absence or no participation will result in a "No Grade" for the project.

Plagiarism is not allowed. If plagiarism is observed, sanctions will follow

4.1 Project Resits

In case an individual, or group, fails the project, the following rules apply. Changes will be permitted to the composition of the original project group after approval from the project examiners. Students who were expelled from the project are not allowed to take the project resit. Next, a resit opportunity will only be offered if the grade was 4 or more. When failing the project, the student(s) will receive an additional assignment by the examiners within 2 working days. This additional assignment has to be handed in within 15 working days.

5. Project coordination

The examiners of project 1-1 are: Pietro Bonizzi, ,Steven Kelk, C. Seiler, and Evgueni Smirnov. Jan Paredis is coordinating the project and tutoring the groups together with Carlo Galuzzi. Every week they will have a meeting with each project group. In block 1 there will be 2 P-meetings during the project week (week 6) For questions and remarks regarding the computers you can mail to lo-fse@maastrichtuniversity.nl. General information, information on the courses and schedules is to be found on the student portal.

Appendices

A **Project evaluation assessment form**

B **Assessment form presentations phase 1 and 2**

C **Criteria for assessing projects**

D **Project groups, tutors and lecturers**

E **Project meetings**

Appendix A Project evaluation assessment form final phase

Date:	Project: 1-1
Student:	ID-number:
Project coordinator: Jan Paredis	Number of the group:
Examiner: C. Seiler	Examiner: Steven Kelk
Examiner: Evgueni Smirnov	
Examiner: Pietro Bonizzi	

Assessment Product	factor x mark	**Score**
Form	1 x	
Contents	2 x	
Product mark		A =

Assessment Report	factor x mark	**Score**
Form	1 x	
Contents	2 x	
Report mark		B =

Assessment Presentation	factor x mark	**Score**
Form	1 x	
Contents	2 x	
Presentation mark		C =

GROUP MARK 1.3	(2A + 2B + C)/5	
FINAL MARK 1-1	(1.1 + 1.2 + 4x1.3)/6	

Attendance P-meetings	Sufficient?	Yes / No

Attendance Project Skills	Sufficient?	Yes / No

Individual mark	
Signature Examiner: Date:	

Appendix B Assessment form presentations phase 1 and 2

Project 1-1: Block 1.1
Group:
Date:

Approach of the assignment: poor / unsatisfactory / average / good / excellent

Is the approach standard / innovative? A single or multiple approach(es)?
Comments:
…………………………………………………………………………………………………

Argumentation of choices: poor / unsatisfactory / average / good / excellent

Are the choices justified? Background literature / theory?
Comments:
…………………………………………………………………………………………………

Results: poor / unsatisfactory / average / good / excellent

Are the results as expected / surprising? Can they be explained? Are the results discussed in a proper way? Are they comparable with each other / literature/ theory?
Comments:
…………………………………………………………………………………………………

Further planning: poor / unsatisfactory / average / good / excellent

Is the planning detailed enough and realistic? Is the role distribution clear and satisfactory?
Comments:
…………………………………………………………………………………………………

Presentation / Demonstration: poor / unsatisfactory / average / good / excellent

Structure and clarity of presentation / demonstration. Answering questions.
Comments:
…………………………………………………………………………………………………

Group collaboration: poor / unsatisfactory / average / good / excellent

How is the collaboration in the group? Changing roles? Comparable contributions?
Comments:
…………………………………………………………………………………………………

Mark phase 1 / Block1 (1-10):

Appendix C Criteria for assessing projects

Product

form

- well-structured, clarity, easy to read
- user-friendly, easy input of data and parameters, good use of graphical possibilities, help facility, robustness (e.g. against deficient input)
- manual, demo and/or other documentation

ccontents

- functionality (has the solution been found?), efficiency (computational time needed)
- range of applicability of the method(s) used
- originality

Report

form

- cover, title page, table of contents, preface
- summary
- list of figures, list of abbreviations and symbols
- problem description
- structure of chapters and paragraphs, use of (sub)titles
- introduction, conlusion
- lay-out and spelling
- references

contents

- summary
- introduction,problem description, report structure
- relevancy and complete reference of sources
- background, history and importance of the problem
- theoretical basis of the method, explanation of the theory used, description of alternative approaches, motivation for the chosen approach
- program design
- workplan
- system validation, description of the test situation, test results, conclusions of the tests, final conclusions
- overview of operational costs
- adequate use of figures, tables, correct en adequate use of mathematics
- explain abstract issues using simple illustrative examples
- coherence of the arguments, well-written for the target audience

Presentation

form

- introduction, title, names of project members, announcement of the structure of the presentation
- contact with the public, audibility, attitude and behaviour
- adequate use of audio-visual support

contents

- problem statement, objectives of the project team, description of the problem
- method of dealing with the problem
- theoretical approach
- explanation of experiments
- results and conclusions
- comprehensibility, quality of the argument(s), correct en adequate use of mathematics

Appendix D Project groups, tutors and lecturers

Here you will find an overview of the different roles and functions of those involved with project centred learning.

Student:
Personal characteristics: autonomous, willing to take initiatives, adaptability, flexibility, capacity for improvisation, willing to cooperate.
Skills: oral presentation techniques, skills to negotiate, social skills; planning, organizing and coordinating activities and competent to apply knowledge.

Tasks:
There are several tasks within a project group.
Project leader:
monitors the process / progress
updates the planning
external communication
responsible for handing in the report and product

Editor:
responsible for the report
checks texts written by other team members
gives writing assignments
puts together the individual contributions resulting in a complete and coherent report
administer of the literature for the report

Presentation co-ordinator:
responsible for the presentation
tests the hardware used for the presentation
ensures coherence of the sheets and slides

Product co-ordinator:
assigns tasks with respect to the creation of the product
links the different modules of the product
manages the documentation, manual(s), diagrams
finishing touch
monitors progress of the product according to the design

Besides the functions above, students are assigned the responsibility for the input of the contents of the courses in the project.

The project groups, in the end, are responsible for the final results and the products. With each project general and specific learning goals can be discerned. The general goals are the same for each project to come. However, each time a higher level is expected. The following general goals are formulated by the educational board:

- Ability to apply knowledge and skills
- Ability to elicit relevant information in an efficient way
- Being able to cooperate in a project group
- Being able to structure meeting
- Being able to deliberate and negotiate with an external client
- Making and adjusting a planning and schedule
- Able to work under time pressure
- Able to give a presentation
- Able to write a project report

The **lecturers** who are responsible for a specific project formulate the level of these more general goals by stating clear criteria.

The more specific goals are different for each project. These goals are content related and related to other educational activities in the block. The specific goals are formulated by those lecturers involved in the block. These goals are for example the application of specific theories, and methods and techniques on behalf of the project.

The role of a tutor is to keep a group on the right track in a non-directive way. He or she observes the process and progress of a group and may give suggestions to encourage the group.

The project coordinator
The project coordinator is responsible for the ins and outs of a project. Every week the coordinator has a meeting with each project group. Besides the project examiner the lecturers, involved in a block, also assess product, report and presentation of each project group.

The lecturers are responsible for the professional knowledge. They will be available for consultation by the students during a project. Since the lecturers may act as a client for a project assignment, it is very important to approach them with your questions and ideas. The lecturers are being expected to question the students during their final presentations on domain related issues.

Appendix E Project meetings

The aim of a project meeting is to continuously track the state of the art of the project by looking backward and forward. Appointments made are checked, new appointments are made. Moreover, the feasibility of the planning will be checked. In case of deviations, an analysis of the situation will be made in order to trace the causes. Project meetings normally are scheduled on a fixed date and time. An agenda is available on each meeting. The chairman and secretary put up the agenda. Of each meeting minutes will be taken.

The agenda below can be seen as a standard agenda. This agenda, of course, can be changed, influenced by the project or specific situation.

1. Opening
2. Announcements
   a. by the group members
   b. by the tutor
3. Incoming/outgoing post or mail
4. Minutes last meeting (mention date of meeting)
5. State of the courses
6. State of the project
   a. planning
   b. correction of the planning
   c. discussion on the log book
   d. discussion on the results of a brainstorm session or of a sub group
7. Cooperation
8. Appointments
   a. tasks to be done
   b. organization of the work
   c. chairman and secretary next meeting; monitoring the log book
9. Any other business

The project leader is chairman. He or she takes care that everybody can participate, that the atmosphere in the group is safe and open, and that conclusions are drawn and decisions are made. Summarizing is an important skill for a chairman.

The secretary takes care of the minutes. He/she sends them at last one day before the next meeting to every group member and the tutor. The minutes have to be provided with date, group number and the names of those present and absent.