

Project Documentation Group 38

JEAN-LOUIS CLEMENTE, ALBINA SHYNKAR, DAVID ELSKAMP*

University of Twente

d.d.w.elskamp@student.utwente.nl, j.l.clemente@student.utwente.nl, a.shynkar@student.utwente.nl

April 16, 2021

According to the following documentation of computational results for the graph isomorphism project, we aim at a grade of 6. Since Jeanlouis put in more work, as evidenced by Table 3, we suggest that all team members receive the group average grade and Jeanlouis receives a +1.0 for his extra effort.¹

Team member	grade (deviation from average)
Albina Shynkar	6.0 (+0.0)
Jeanlouis Clemente	6.0 (+0.0)
David Elskamp	6.0 (+0.0)

Table 1: Suggested grades

1. CATEGORY 6: BASIC PROBLEM INSTANCES

The basic problem instance have been solved correctly during the programming competition. The following table shows the computation times. (The “X” should not be there when you want to have a sufficient grade ;-)

instance	correctly solved	comp. time (s)
basicGI1	✓	0.7
basicGI2	✓	40.2
basicGI3	✓	3.4
basicGIAut	✓	16.8
basicAut1	✓	43.2
basicAut2	✓	5.6

Table 2: Computation times basic instances.

2. CATEGORY +1: ADDITIONAL TECHNIQUES FOR FASTER ALGORITHMS

Not implemented.

*Fourth group member Boris Belchev was removed from our team after 1 week.

¹If you choose a non-uniform distribution, let the grades in the table reflect the suggested final grade of the team member with in between brackets the deviation. So if the average is 6.0 and Jeanlouis put in extra effort, his grade could be 6.0 (+1.0). A non-uniform distribution should also be reflected in the team-building contract and/or the Buddycheck evaluation, to give everyone a chance to change this outcome.

3. CATEGORY +1: IMPLEMENTATION OF FAST PARTITION REFINEMENT

Not implemented.

4. CATEGORY +1: USING GENERATING SETS FOR COMPUTING $| \text{Aut}(G) |$

Not implemented.

5. CATEGORY +1: ADDITIONAL AND GENUINELY NEW IDEAS

As a Group we had some issues with color refinement initially and when we finally got to solid implementation, we ran into more bugs specifically when we implemented the the branching algorithm. This lead to a lot of wasted time and acted as hindrance to meeting the minimum requirements. However, we manged to rewrite the code and this time around it worked perfectly.

In terms of genuinely new ideas, as a group we came up with a few promising ideas, however none of these ideas really panned out. For instance we tried implementing color refinement non-recursively by using a pointer to activate certain parts of the code when needed. The main reason we choose to do this was to avoid recursion limit issues, however the code itself was really large and hard to follow so debugging any issues began very time consuming and as a result we abandoned this idea.

6. BALANCE SHEET & REFLECTION

6.1. Work Distribution

The following is the estimated work distribution in the group for this implementation project.

	A. Shynkar	J. Celemente	D. Elskamp
Color Refinement	40%	50%	10%
Branching Algorithm	20%	50%	30%
Preprocessing Trees	–	–	–
Preprocessing Twins	–	–	–
Fast Part. Refinement	–	–	–
Nonsense Count	–	–	–

Table 3: *Work Distribution Group 38.*

6.2. Team Dynamics

We ask you to briefly comment on both positive and/or negative experiences with respect to teamwork and team composition in your project group, specifically with respect to the interdisciplinary team composition (if applicable). You can also include your experience with the SDL Teamwork Development and Buddycheck. Try to be short, but as specific as possible.

Examples could include:

- + Communication was good via Whatsapp and google meets and everybody was always on time.
- + With the small team and little time because of double modules we still managed to get enough work done.
- As we were all very busy and Jeanlouis and Albina were doing a double module we sometimes had a shortage of time and possible friction within the group dynamics.
- Initially the communication and organization was a little chaotic.