

Saucy_Basic

Ahmed Jazzar¹ Bassam Jaber²
Prof. Karem Sakallah³

¹University of Jordan

²Birzeit University

³University of Michigan

2013

Outline

- 1 Background and problem definition
 - Background
 - Applications
 - I/O
- 2 Approach used
 - The Approach
 - Challenges and obstacles
 - Division of work
 - programming language
 - References
- 3 Results
 - OutComes
 - Tests
- 4 Lessons Learned
 - Lessons Learned
- 5 Thanks

Outline

- 1 Background and problem definition
 - Background
 - Applications
 - I/O
- 2 Approach used
 - The Approach
 - Challenges and obstacles
 - Division of work
 - programming language
 - References
- 3 Results
 - OutComes
 - Tests
- 4 Lessons Learned
 - Lessons Learned
- 5 Thanks

Background and problem definition

Background

- Graph Symmetries
 - Graph Symmetry is a permutation of some vertices in a graph that leaves the graph unchanged.

Background and problem definition

Background

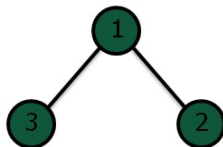
- Graph Symmetries

- Graph Symmetry is a permutation of some vertices in a graph that leaves the graph unchanged.

Original Edges

1 - 2

1 - 3



Background and problem definition

Background

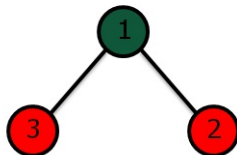
- Graph Symmetries

- Graph Symmetry is a permutation of some vertices in a graph that leaves the graph unchanged.

Original Edges

1 - 2

1 - 3



Background and problem definition

Background

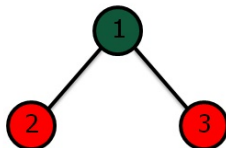
- Graph Symmetries

- Graph Symmetry is a permutation of some vertices in a graph that leaves the graph unchanged.

Original Edges

1 - 2

1 - 3



New Edges

1 - 3

1 - 2

Background and problem definition

Background

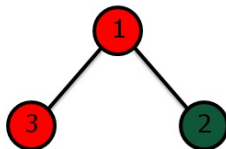
- Graph Symmetries

- Graph Symmetry is a permutation of some vertices in a graph that leaves the graph unchanged.

Original Edges

1 - 2

1 - 3



New Edges

Background and problem definition

Background

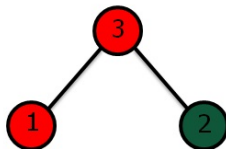
- Graph Symmetries

- Graph Symmetry is a permutation of some vertices in a graph that leaves the graph unchanged.

Original Edges

1 - 2

1 - 3



New Edges

3 - 1

3 - 2

Background and problem definition

Background

- Graph Symmetries

- Graph Symmetry is a permutation of some vertices in a graph that leaves the graph unchanged.
- Number of possible symmetries are $n!$, n is the number of vertices. So in the previous examples we have $3!$ Possible symmetries.

Background and problem definition

Background

- Graph Symmetries

- Graph Symmetry is a permutation of some vertices in a graph that leaves the graph unchanged.
- Number of possible symmetries are $n!$, n is the number of vertices. So in the previous examples we have $3!$ Possible symmetries.
- Our problem is to find the permutations which leaves the graph unchanged, from a given graph with n vertices and number of partitions(colors) of its vertices.

Outline

- 1 Background and problem definition
 - Background
 - **Applications**
 - I/O
- 2 Approach used
 - The Approach
 - Challenges and obstacles
 - Division of work
 - programming language
 - References
- 3 Results
 - OutComes
 - Tests
- 4 Lessons Learned
 - Lessons Learned
- 5 Thanks

Background and problem definition

Applications

- Graph Symmetries can be used in different graph applications:
 - Modeling of chemical compounds.
 - Representing migration path or movement between regions.
 - modeling transport networks.
 - logistic optimization.
 - Symmetry Breaking.

Outline

- 1 Background and problem definition
 - Background
 - Applications
 - I/O
- 2 Approach used
 - The Approach
 - Challenges and obstacles
 - Division of work
 - programming language
 - References
- 3 Results
 - OutComes
 - Tests
- 4 Lessons Learned
 - Lessons Learned
- 5 Thanks

Background and problem definition

I/O

- The Input of the program is a text file with the following structure:

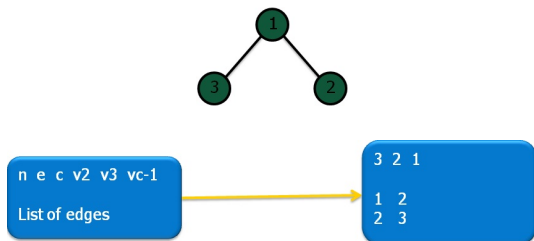
```
n e c v2 v3 vc-1
```

List of edges

Background and problem definition

I/O

- The Input of the program is a text file with the following structure:



Background and problem definition

I/O

- The Input of the program is a text file
- The Output is a list of vertices permutations that leaves the graph unchanged, symmetries number, dot file contains the graph in dot language, and a text file contains some statistics of the graph.

Outline

- 1 Background and problem definition
 - Background
 - Applications
 - I/O
- 2 Approach used
 - **The Approach**
 - Challenges and obstacles
 - Division of work
 - programming language
 - References
- 3 Results
 - OutComes
 - Tests
- 4 Lessons Learned
 - Lessons Learned
- 5 Thanks

Approach used

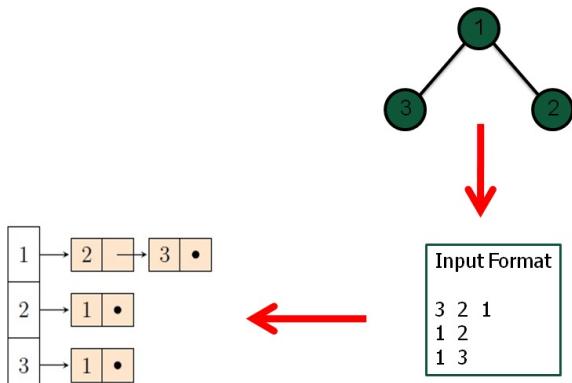
The Approach

- Input Parsing
 - Reading input file
 - Implementing the graph structure

Approach used

The Approach

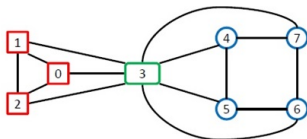
- Input Parsing
 - Reading input file
 - Implementing the graph structure



Approach used

The Approach

- Input Parsing
 - Reading input file
 - Implementing the graph structure
 - Implementing the OP.

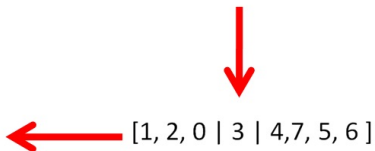


Position:	0	1	2	3	4	5	6	7
Vertex[i]	1	2	0	3	4	7	5	6

Vertex v	0	1	2	3	4	5	6	7
Position[v]	2	0	1	3	4	6	7	5

Position:	0	1	2	3	4	5	6	7
Vertex[i]	1	2	0	3	4	7	5	6

Position:	0	1	2	3	4	5	6	7
Vertex[i]	2	-	0	3	-	-	-	-



Approach used

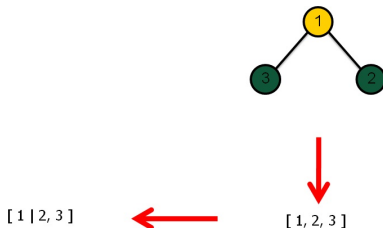
The Approach

- Input Parsing
 - Reading input file
 - Implementing the graph structure
 - Implementing the OP.
- Partition Refinement

Approach used

The Approach

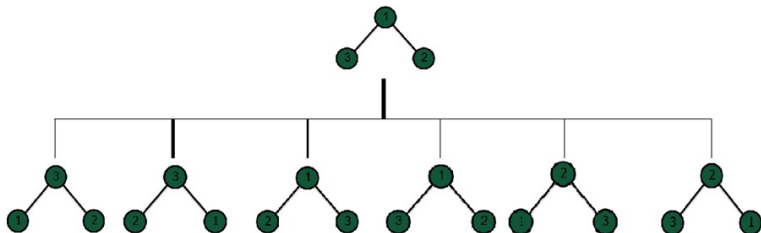
- Input Parsing
 - Reading input file
 - Implementing the graph structure
 - Implementing the OP.
- Partition Refinement



Approach used

The Approach

- Input Parsing
 - Reading input file
 - Implementing the graph structure
 - Implementing the OP.
- Partition Refinement
- Basic Search Tree



Outline

- 1 Background and problem definition
 - Background
 - Applications
 - I/O
- 2 Approach used
 - The Approach
 - **Challenges and obstacles**
 - Division of work
 - programming language
 - References
- 3 Results
 - OutComes
 - Tests
- 4 Lessons Learned
 - Lessons Learned
- 5 Thanks

Approach used

Challenges and obstacles

- Choosing an appropriate data structure.
- Coding a non-time consumer code.

Outline

- 1 Background and problem definition
 - Background
 - Applications
 - I/O
- 2 Approach used
 - The Approach
 - Challenges and obstacles
 - **Division of work**
 - programming language
 - References
- 3 Results
 - OutComes
 - Tests
- 4 Lessons Learned
 - Lessons Learned
- 5 Thanks

Approach used

Division of work

- Before dividing the problem, we understand it.
- Each one of us take a part of the problem and solve it individually.
- After finishing our tasks we combine the works each to other and check the final result in order to send it to our supervisor for his feedback.

Outline

- 1 Background and problem definition
 - Background
 - Applications
 - I/O
- 2 Approach used
 - The Approach
 - Challenges and obstacles
 - Division of work
 - **programming language**
 - References
- 3 Results
 - OutComes
 - Tests
- 4 Lessons Learned
 - Lessons Learned
- 5 Thanks

Approach used

programming language

- Saucy_Basic programmed in C language.
- Its also developed in Code::Blocks at windows, and Ubuntu environments

Outline

- 1 Background and problem definition
 - Background
 - Applications
 - I/O
- 2 Approach used
 - The Approach
 - Challenges and obstacles
 - Division of work
 - programming language
 - **References**
- 3 Results
 - OutComes
 - Tests
- 4 Lessons Learned
 - Lessons Learned
- 5 Thanks

Approach used

References

- We return to the project description every time we feel that we dont understand the problem completely.
- If the project description didnt give us what we want, we refer to the materials given by our supervisor.
- Our Supervisor is the most important reference we refer for.

Outline

- 1 Background and problem definition
 - Background
 - Applications
 - I/O
- 2 Approach used
 - The Approach
 - Challenges and obstacles
 - Division of work
 - programming language
 - References
- 3 Results
 - **OutComes**
 - Tests
- 4 Lessons Learned
 - Lessons Learned
- 5 Thanks

Results

OutComes

```

C:\Windows\System32\cmd.exe
C:\Users\g.j\Desktop\Documentation\file\bin\Release>C:\Users\g.j\Desktop\Documentation\file\bin\Release\test.exe InputFile.txt OutFile.txt Graph.h
Reading Time :0.000000
File Read Done.
Go And Check OutFile.txt File For OutPut
writing Time :0.005000
Write Graph VIZ Done
Create Initial OP for the Graph

Start Refinement of the OP

Top
| 1 2 | 0
Bottom
| 1 2 | 0

Refinement Time :0.001000
Start finding Symmetries

1 1 2 | 0
1 1 2 | 0

1 1 2 | 0
2 1 1 | 0

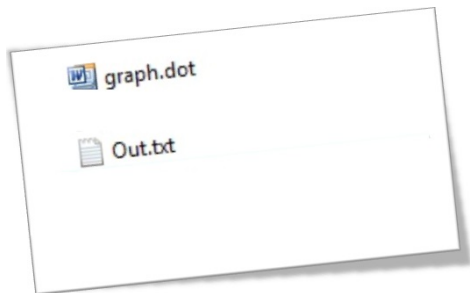
Symmetries found in Time :0.001000
Symmetries Found 2
C:\Users\g.j\Desktop\Documentation\file\bin\Release>

```

Output Screen

Results

OutComes



Out Files

Outline

- 1 Background and problem definition
 - Background
 - Applications
 - I/O
- 2 Approach used
 - The Approach
 - Challenges and obstacles
 - Division of work
 - programming language
 - References
- 3 Results
 - OutComes
 - **Tests**
- 4 Lessons Learned
 - Lessons Learned
- 5 Thanks

Results

Tests

- Before we deliver every part of our project, the test was being by comparing the results with the correct results provided.

Results

Tests

- Before we deliver every part of our project, the test was being by comparing the results with the correct results provided.
- These data(input, and its correct output) are provided in Project Description, and some slides given by our supervisor.

Results

Tests

- Before we deliver every part of our project, the test was being by comparing the results with the correct results provided.
- These data(input, and its correct output) are provided in Project Description, and some slides given by our supervisor.
- After that the final testing has been done by prof. Karems team in the University of Michigan, and he notified us about every small issue in our deliverables.

Outline

- 1 Background and problem definition
 - Background
 - Applications
 - I/O
- 2 Approach used
 - The Approach
 - Challenges and obstacles
 - Division of work
 - programming language
 - References
- 3 Results
 - OutComes
 - Tests
- 4 **Lessons Learned**
 - **Lessons Learned**
- 5 Thanks

Lessons Learned

Main things

- Organizing My time.
- Working in a team.
- Learn myself.
- Overcome the problems.

Lessons Learned

I wish that I knew before the project

- Some practice with teams.

Lessons Learned

If I do it again

- Complete it quickly to start doing Symmetry Breaking.

Lessons Learned

Advices

Its easy, if you believe in yourselves!

Thank you!