Automated Exercise Generation for an Algorithmics Course

Zoltan Sojtory

What is *Automated Exercise Generation*?

- Producing many distinct variants of an exercise
- Three main areas
  - Generating new problems
  - Generating solutions
  - Automatic grading
- Each have their own difficulties

Motivation / Previous Works

- Online Examinations
- Previous papers
  - Embedded systems MOOCs - D. Sadigh, S. A. Seshia, and M. Gupta. Automating Exercise Generation: A Step towards Meeting the MOOC Challenge for Embedded Systems
  - Satisfiability problems - P. Hozzová, L. Kovács, and J. Rath. Automated Generation of Exam Sheets for Automated Deduction
  - Data Science - C. Kotsiopoulos, I. Doudoumis, P. Raftopoulou, and C. Tryfonopoulos. DaST: An Online Platform for Automated Exercise Generation and Solving in the Data Science Domain
- None discuss graph/string algorithms from Algorithmics Courses

Chosen algorithms

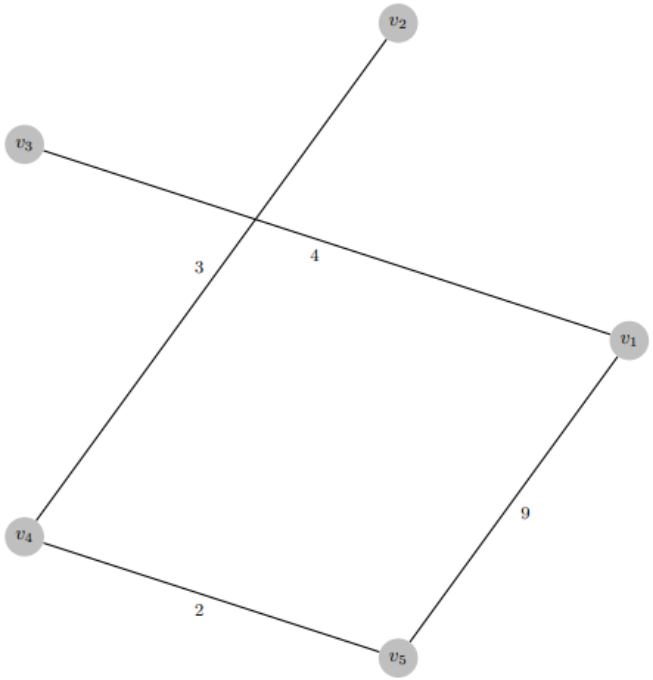
- Focus on problem and solution generation
- Dijkstra's Algorithm (shortest path)
  - The exercises generated ask students to identify the shortest paths from a given a vertex to all other vertices in a weighted graph.
- KMP Algorithm (string search)
  - For this exercise, students are required to build the border table of a given string, so that the KMP algorithm can be applied to search for a string.

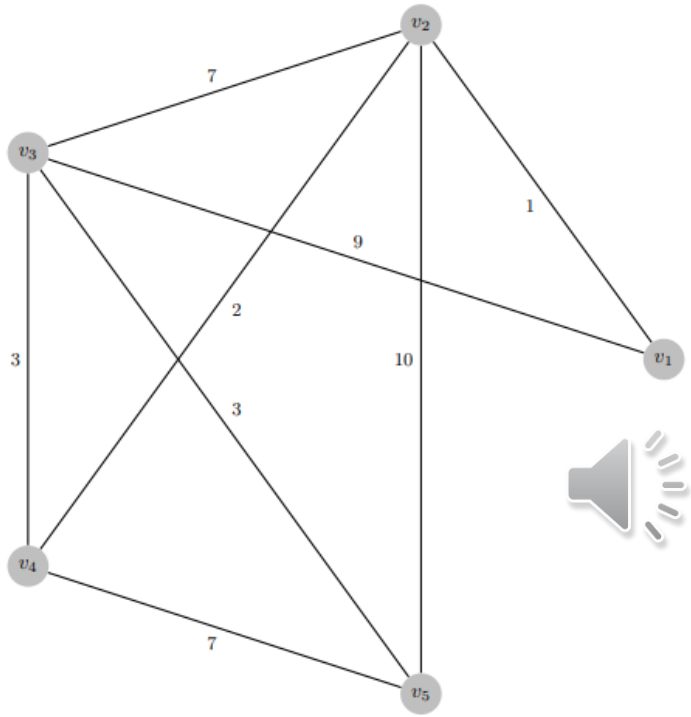
Approach / Difficulties

- Challenges with problem generation
  - Ensuring uniform difficulty
- Template Based Approach
  - By reviewing existing, handwritten problems
  - Identify characteristics which alter difficulty

Dijkstra Template

- Number of vertices (User input)
- Number of edge relaxations (User input)
- Number of edges
- Edge weights





KMP Template

- String size (User input)

- Longest border size (User input)
- Overlapping or Non-overlapping longest border (User input)
  - “ababa……” (Overlapping) vs “aba…….aba….” (Non-overlapping)

- Alphabet

Dijkstra Generation Algorithm

1. Generate the lengths of the shortest paths to each vertex from the starting vertex.
2. Insert vertices into the graph by connecting them based on their shortest distances.
3. Insert additional edges for edge relaxations, creating the final graph.

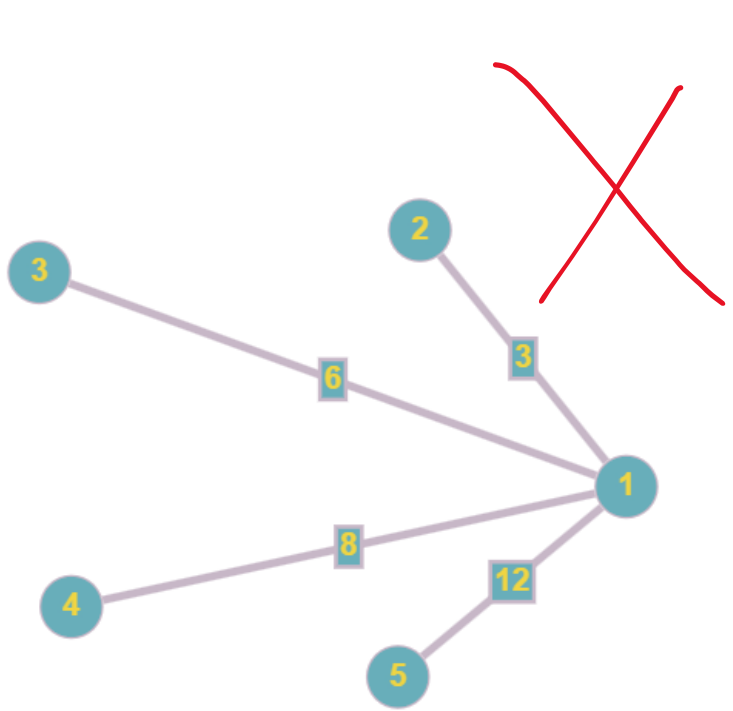
5 vertices, 3 edge relaxations example

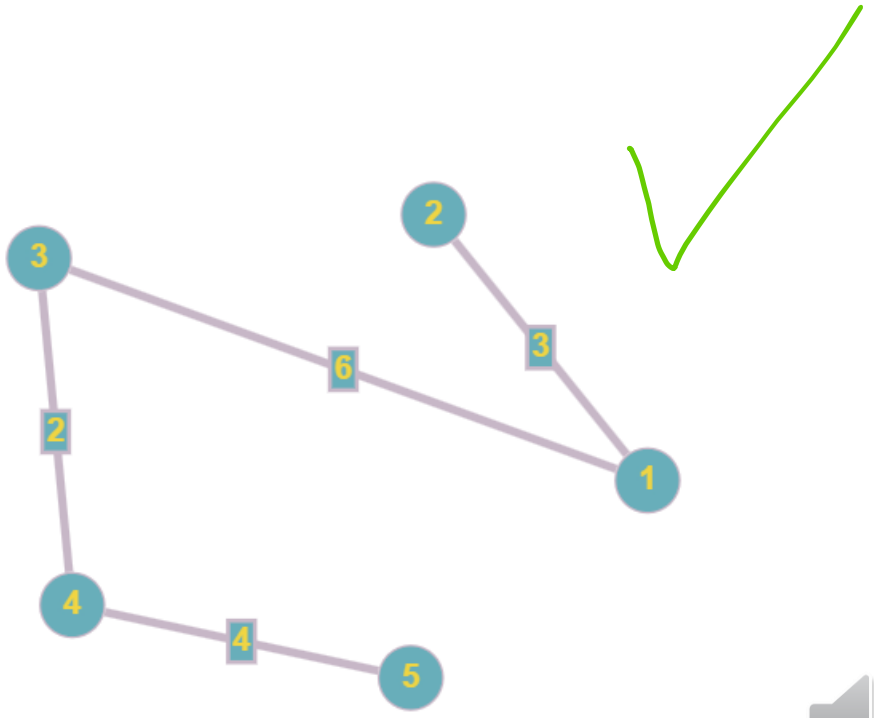
1.

V1	V2	V3	V4	V5
0	3	6	8	12

5 vertices, 3 edge relaxations example

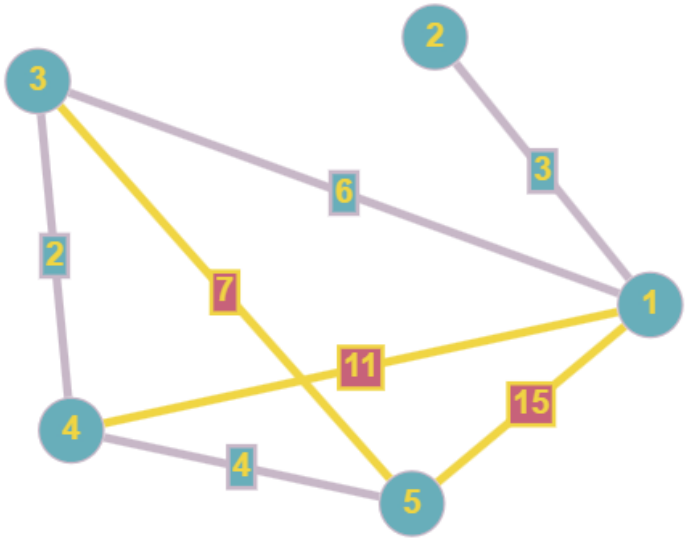
2. Insertion must be dynamic.





5 vertices, 3 edge relaxations example

3.



KMP Generation Algorithm

1. Generate the largest border.
2. Fill out the rest of the string.

13 length, 4 length longest non-overlapping border example

1. [_, _, _, _, _, _, _, _, _, _, _, _, _, _] (alphabet: {C, G, T})

Randomly generated largest border: CGGT

Inserted at random location (and beginning):

[C, G, G, T, _, _, _, _, C, G, G, T, _]

Ensure subsequent characters do not match:

[C, G, G, T, C, _, _, _, C, G, G, T, C] ✗

[C, G, G, T, C, _, _, _, C, G, G, T, T] ✓

13 length, 4 length longest non-overlapping border example

2. Fill in empty spaces with random characters from alphabet:

[C, G, G, T, C, G, G, T, C, G, G, T, T]

Ensure no repetitions of longest border:

[C, G, G, T, C, G, G, T, C, G, G, T, T] → [C, G, G, T, G, G, G, T, C, G, G, T, T]

Implementation

- Java object oriented approach
- LaTeX generation
- PDF rendering

Evaluation

1. How similar are generated exercises/solutions to pre-existing example exercises/solutions of the same type?

2. What are some of the ethical implications of automatic exercise generation?

3. How good is the usability of the application?

4. To what extent are controlled and uncontrolled variables handled effectively?

Conclusion

- Summary
- Future work
  - In-depth proofs
  - Further studies

Sources

G. Eshiba-Emir. Automated exercise generation for three satisfiability checking algorithms, 2022.

P. Hozzová, L. Kovács, and J. Rath. Automated Generation of Exam Sheets for Automated Deduction. PhD thesis, TU Wien, 2021.

C. Kotsiopoulos, I. Doudoumis, P. Raftopoulou, and C. Tryfonopoulos. DaST: An Online Platform for Automated Exercise Generation and Solving in the Data Science Domain. PhD thesis, University of Peloponnese, 2019.

D. Sadigh, S. A. Seshia, and M. Gupta. Automating Exercise Generation: A Step towards Meeting the MOOC Challenge for Embedded Systems. PhD thesis, UC Berkeley, 2012.