CS 291 Mini Project

Presentation

Lars Yarkosky, Abby Gerstner, Semyon Tsyrenov

Introduction

Description of Project

Description:
1. Provide a list of courses and their prerequisites in a fictional academic program.
2. Create a graph where courses are nodes, and directed edges represent prerequisites.
3. Allow for cycles to show that some courses may have multiple prerequisites.

Objectives and Goals

- Main objective: To model course dependencies using a directed graph.
- Identify any courses that can be taken without prerequisites
- Find the courses with the most and least prerequisites
- Determine if there are any circular dependencies
- Create a simple and easy-to-use interface

Problem Statement

Define the Problem

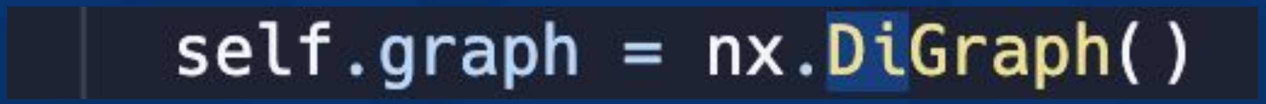
Our project addresses the problem of analyzing and managing academic course dependencies within a program, offering a user-friendly interface to input course prerequisites, identify courses without prerequisites, find those with the most or least prerequisites, and detect potential circular dependencies, providing valuable insights for academic program administrators or students.

Why is it important to address?

Analyzing and managing academic course dependencies is important for educational institutions and students to ensure efficient curriculum planning, timely completion of academic requirements, and the avoidance of scheduling conflicts.

Relevant data

According to this reddit thread, many students feel extremely stressed out by registration, with this program, some of that stress could be reduced as it minimizes needing to keep track of all course prerequisites.



```
self.graph = nx.DiGraph()
```

Methodology/ Implementation

Research methodology

- Used notes from this class about graphs, directed graphs, and cycles
- Researched Tkinter library and Networkx library to use within our Python code
- Tkinter library: the standard GUI library for Python
- Networkx library: Python library for studying graphs and networks

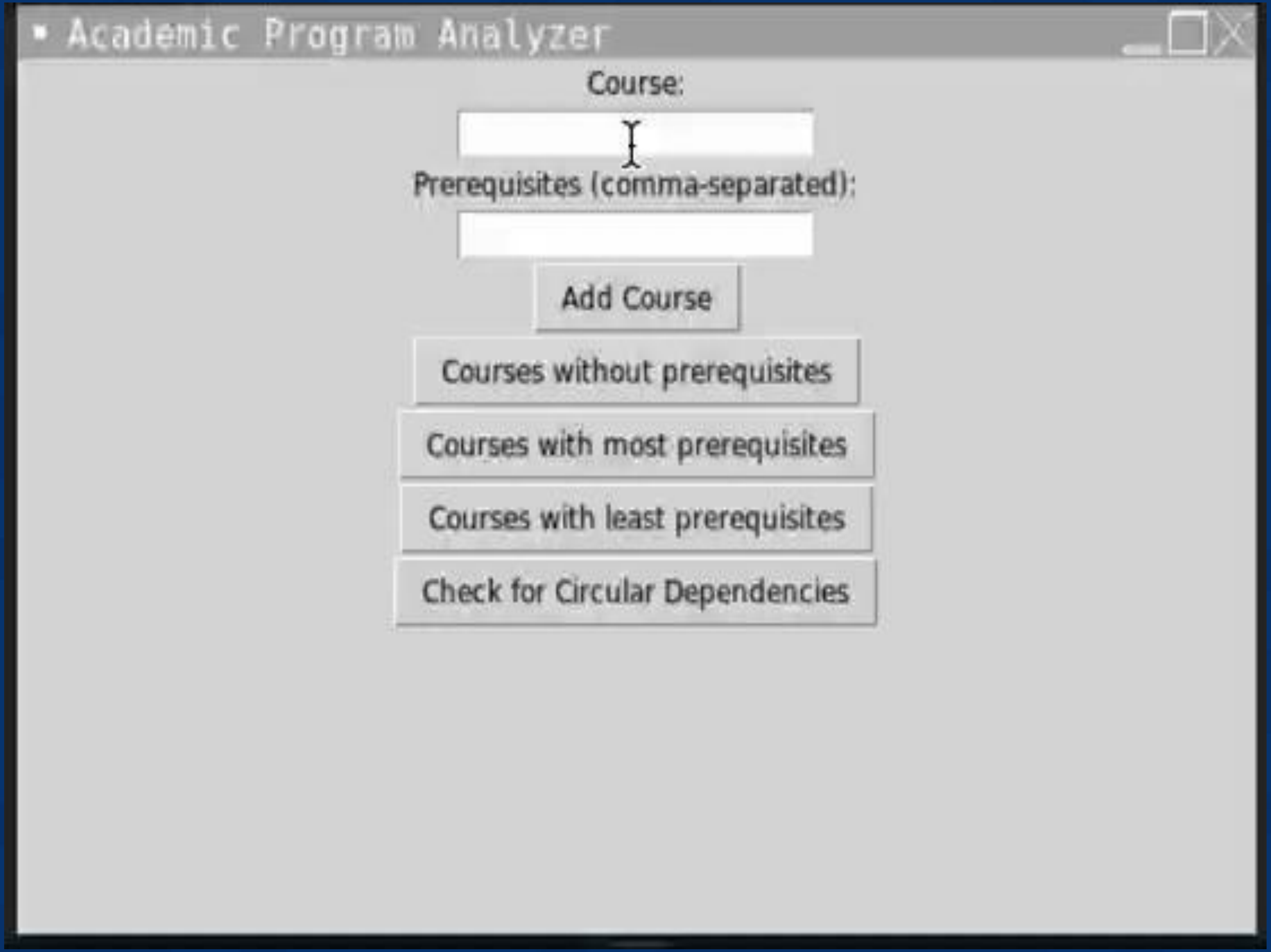
Tools used

- Repl.it to collaborate on code
- Tkinter library
- Networkx library

Implementation

- Implemented digraphs and graph tools into Python code using Networkx
- Add course function: adds course as a new node of the graph and for each prerequisite, it adds one edge
- Courses without prerequisites function: checks to see which nodes have no edges, which represent prerequisites
- Courses with most prerequisites function: checks to see which node/s has the most edges
- Courses with least prerequisites function: checks to see which node/s have the least edges
- Check for circular dependencies function: Makes sure that no nodes are dependent upon each other

Results


Academic Program Analyzer

Course:

Prerequisites (comma-separated):

Add Course
Courses without prerequisites
Courses with most prerequisites
Courses with least prerequisites
Check for Circular Dependencies

Discussion

Analyze the results

- The program handles all class entries/ prerequisite entries in an elegant and simple way. This event driven program works as was intended, and consistently gives accurate results.

Compare with Initial Objectives

- Our initial objective was to make a GUI to simplify class registration. We believe our model does a good job of handling the issues faced by students at registration time.

Limitations

- A limitation for this project would be that users have to manually enter the courses and the courses prerequisites. Creating a model that scrapes the UMKC website would be much more efficient for the user.

Conclusion

In summary, this application simplifies the class registration process by tracking which courses students can take depending on which prerequisites they have already taken.

Future Work

Timeline Planning:

Implement a timeline planning feature that allows users to schedule courses over multiple semesters, considering prerequisites and dependencies.

References

- https://www.reddit.com/r/college/comments/tt9rlr/why_is_signing_up_for_classes_the_most_stressful/
- https://chat.openai.com/
- https://networkx.org/documentation/stable/reference/classes/digraph.html
- https://docs.python.org/3/library/tkinter.html

Thank you!

Questions?