CSCE 221 Cover Page

Programming Assignment #5

First Name Dilanka Last Name Weerasinghe UIN 126007816 User Name dweerasinghe E-mail address dweerasinghe@tamu.edu

Please list all sources in the table below including web pages which you used to solve or implement the current homework. If you fail to cite sources you can get a lower number of points or even zero. According to the University Regulations, Section 42, scholastic dishonesty are including: acquiring answers from any unauthorized source, working with another person when not specifically permitted, observing the work of other students during any exam, providing answers when not specifically authorized to do so, informing any person of the contents of an exam prior to the exam, and failing to credit sources used. Disciplinary actions range from grade penalties to expulsion read more: Aggie Honor System Office

Type of sources			
People			
Web pages (provide URL)	https://www.youtube.com/watch?v=_LuIvEi_kZk&t=306s		
Printed material			
Other Sources			

I certify that I have listed all the sources that I used to develop the solutions/codes to the submitted work.

"On my honor as an Aggie, I have neither given nor received any unauthorized help on this academic work."

Electronic signature Dilanka Weerasinghe Date 4/27/20

Programming Assignment 5 (100 points) – due April 27

- **Part 3 (20 points) – due April 27**
  - Why does the algorithm use a queue? Can we use a stack instead?
    - You can use a stack instead of a queue. You are able to use any collection based data structure as long as you can add an element to it and remove some element form it. It is not required to be FIFO. This does require you to change the topological sorting algorithm to fit the collection structure. The queue is used in this implementation of topological sort because it will already send a vertex with an indegree that can be decremented in the right order. The order of decrementation is what matters not the vertex that goes in that is why you can hold the vertex with anything.
  - Can you explain why the algorithm detects cycles?
    - You start at an arbitray vertex with no incoming edges and you traverse throught the graph indegrees. If you continue to use this algorithym to decrement the indegrees the counter will fail because it will count more indegrees than there are remaining in throughout the vectors becasue there will be a cycle if the indegrees counted does not match up to the indegrees being removed.
  - What is the running time for each function? Use the Big-O notation asymptotic notation and justify your answer.
    - The running time of the topological sort is O(V+E) where V are the vertex and E are the egdes.
      - This is because the algoritym runs through the vertex once then it will queue up each corresponing edge for the indegree decementation. Hence, E + V
    - The running time of my compute indegree is O(V*E*E).
      - This is because to find the indegree I first have to iterate through the vectors -> V. Within that iteration I have to iterate through the edges of each of the vecors the increment the indegree for the first iteration of vectors -> E^2 . This means that it must be V*E^2.
  - (5 points) test your program for correctness using the four cases below:

Case 1: Use the example (`input.data`) provided in the description of the problem.

- output
  - Display Graph
    - 1: 2 4 5
    - 2: 3 4 7
    - 3: 4
    - 4: 6 7
    - 5: 6: 5 7: 6
    - 1 2 3 4 7 6 5
    - This is the correct topological sort

Case 2: Samantha plans her course schedule. She is interested in the following eight courses: CSCE121, CSCE222, CSCE221, CSCE312, CSCE314, CSCE313, CSCE315, and CSCE411. The course prerequisites are:

course	#	prerequisites	
CSCE121:	1	(none)	
CSCE222:	2	(none)	
CSCE221:	3	CSCE121	CSCE222
CSCE312:	4	CSCE221	
CSCE314:	5	CSCE221	
CSCE313:	6	CSCE221	
CSCE315:	7	CSCE312	CSCE314
CSCE411:	8	CSCE222	CSCE221

- output
- Display Graph
- 1: 3
- 2: 3 8
- 3: 4 5 6 8
- 4: 7
- 5: 7
- 6:
- 7:
- 8:
- 1 2 3 4 5 6 8 7
- The sort of the order of classes she can take is ...
- CSCE121 - CSCE222 - CSCE221 - CSCE312 - CSCE314 - CSCE313 - CSCE315 - CSCE411

Find a sequence of courses that allows Samantha to satisfy all the prerequisites. Assume that she can only take one class at a time. The input file for this case is provided (`input2.data`)

Case 3: Samantha loves foreign languages and wants to plan her course schedule. She is interested in the following nine courses: LA15, LA16, LA22, LA31, LA32, LA126, LA127, LA141, and LA169. The course prerequisite are:

course	#	prerequisites	
LA15:	1	(none)	
LA16:	2	LA15	
LA22:	3	(none)	
LA31:	4	LA15	
LA32:	5	LA16	LA31
LA126:	6	LA22	LA32
LA127:	7	LA16	
LA141:	8	LA22	LA16
LA169:	9	LA32	

- output
- Display Graph
- 1: 2 4
- 2: 5 7 8
- 3: 6 8
- 4: 5 5
- 5: 6 9
- 6:
- 7:
- 8:
- 9:
- 1 3 2 4 7 8 5 6 9
- The order of languages she can take is...
- LA15 - LA22 - LA16 - LA31 - LA127 - LA141 - LA32 - LA126 - LA169

Find a sequence of courses that allows Samantha to satisfy all the prerequisites. Assume that she can only take one class at a time.

Case 4. Create a directed graph with cycles and test your program. There is one such a file provided (`input-cycle.data`).

- If you use a graph with cycles the topological sort function will fail becuase the counter is thrown off. In this case we used exception handling to solve the issue. output
- Display Graph
- 1: 2 4 5
- 2: 3 4 7
- 3: 4
- 4: 6 7
- 5: 4
- 6: 5
- 7: 6
- Cycle found exception