CSE 211: Discrete Mathematics **(Due: 04/01/21)**

Homework #3

Instructor: Dr. Zafeirakis Zafeirakopoulos
Assistant: Gizem Süngü

Course Policy: Read all the instructions below carefully before you start working on the assignment, and before you make a submission.

- It is not a group homework. Do not share your answers to anyone in any circumstance. Any cheating means at least -100 for both sides.
- Do not take any information from Internet.
- No late homework will be accepted.
- For any questions about the homework, send an email to gizemsungu@gtu.edu.tr
- The homeworks (both latex and pdf files in a zip file) will be submitted into the course page of Moodle.
- The latex, pdf and zip files of the homeworks should be saved as "Name_Surname_StudentId".{tex, pdf, zip}.
- If the answers of the homeworks have only calculations without any formula or any explanation -when needed- will get zero.
- Writing the homeworks on Latex is strongly suggested. However, hand-written paper is still accepted **IFF** hand writing of the student is clear and understandable to read, and the paper is well-organized. Otherwise, the assistant cannot grade the student's homework.

Problem 1: Representing Graphs (10 points)

Represent the graph in Figure 1 with an adjacency matrix. Explain your representation clearly.

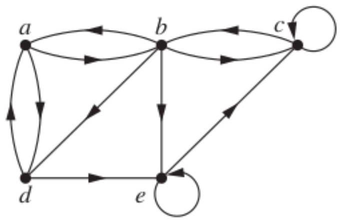


Figure 1: The graph for Problem 1

(Solution)

- Vertex: a, b, c, d, e
- Edge: a→b, a→d, b→a, b→c, b→d, b→e, c→b, c→c, d→a, d→e, e→c, e→e
- First, vertexes are written to the rows and columns in the table we will create.

	a	b	c	d	e
a					
b					
c					
d					
e					

- The number of edges for each (row, column) pair is written in the corresponding part.

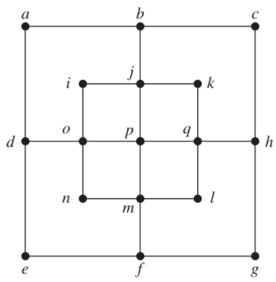
	a	b	c	d	e
a	0	1	0	1	0
b	1	0	1	1	1
c	0	1	1	0	0
d	1	0	0	0	1
e	0	0	1	0	1

- The resulting table is an adjacency matrix.

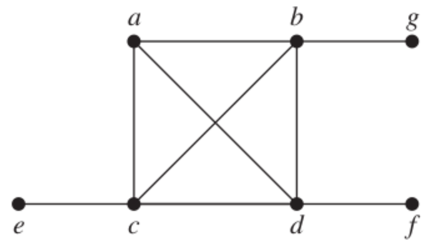
$$\begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Problem 2: Hamilton Circuits (10+10+10=30 points)

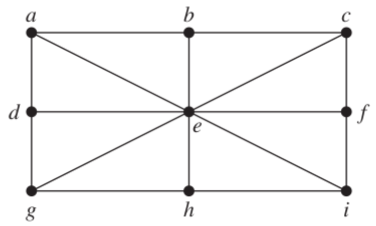
Determine whether there is a Hamilton circuit for each given graph (See Figure 2a, Figure 2b, Figure 2c). If the graph has a Hamilton circuit, show the path with its vertices which gives a Hamilton circuit. If it does not, explain why no Hamilton circuit exists.



(a) The graph G_1



(b) The graph G_2



(c) The graph G_3

Figure 2: The graphs to find Hamilton circuits for Problem 1

(Solution)

Properties of Hamilton circuit

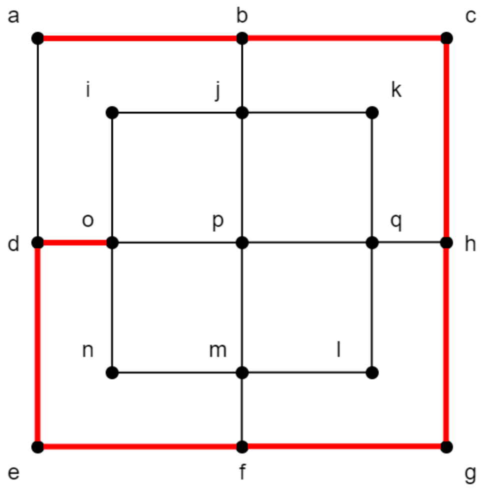
- Returns the starting point
- Visits every vertex once and no repeats

Dirac's Theorem- "If G is a simple graph with n vertices with n$\geq$ 3 such that the degree of every vertex in G is at least n/2, then G has a Hamiltonian circuit."

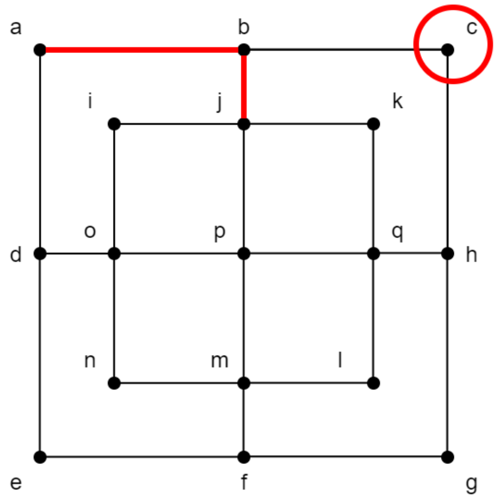
I will first apply the Dirac's Theorem for each graph.

(a)
(Solution)

- Number of vertex= 17
- Degree of vertex = 17/2 = 8,5
- There are vertexes less than 8,5 degrees. Because it is lower grade vertex, we cannot call this circuit definite Hamilton circuit. Therefore, we will try to find the Hamilton circuit.
- When I start with any external node (a, b, c, d, h, e, f, g), I can come back to where I started when I visit all the outside nodes. Otherwise, it is not possible for me to go back to where I started. I cannot visit an outside node in case I visit the nodes inside. If I visit all the outside nodes, I cannot visit the nodes inside.



(a) When start node a and visits all outside nodes



(b) When start node a and visits inside nodes

- When I start with any node (i, j, k, o, p, q, n, m, l) inside, when I need to visit the outside nodes, I still cannot visit an outside node.

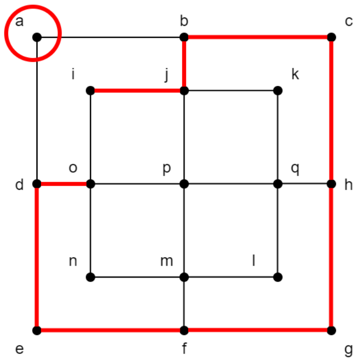


Figure 4: When start node i and visits all outside nodes

- Therefore, Hamilton circuit does not exist in this graph.

(b)
(Solution)

- Number of vertex= 7
- Degree of vertex = 7/2 = 3,5
- There are vertexes less than 3,5 degrees. Because it is lower grade vertex, we cannot call this circuit definite Hamilton circuit. Therefore, we will try to find the Hamilton circuit.
- Since e, g and f vertex have one edge, I cannot go back after starting or visiting them. Because each vertex must be visited only once in the Hamilton circuit. For this reason, there is no Hamilton circuit in this graph.

(c)
(Solution)

- Number of vertex= 9
- Degree of vertex = 9/2 = 4,5
- There are vertexes less than 4,5 degrees. Because it is lower grade vertex, we cannot call this circuit definite Hamilton circuit. Therefore, we will try to find the Hamilton circuit.
- There are Hamilton circuit:

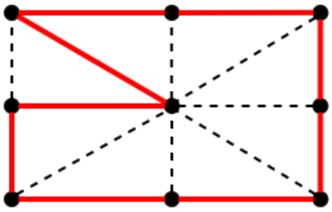


Figure 5: The Hamilton circuit for Problem 2.c

Problem 3: Applications on Graphs (20 points)

Schedule the final exams for Math 101, Math 243, CSE 333, CSE 346, CSE 101, CSE 102, CSE 273, and CSE 211, using the fewest number of different time slots, if there are no students who are taking:

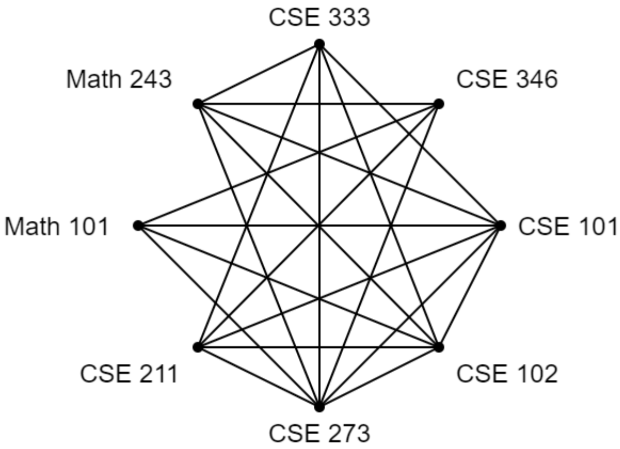
- both Math 101 and CS 211,
- both Math 243 and CS 211,
- both CSE 346 and CSE 101,
- both CSE 346 and CSE 102,
- both Math 101 and Math 243,
- both Math 101 and CSE 333,
- both CSE 333 and CSE 346

but there are students in every other pair of courses together for this semester.
Note: Assume that you have only one classroom.

Hint 1: Solve the problem with respect to your problem session notes.
Hint 2: Check the website

(Solution)

- Firstly i created graph with given informations. Courses not taken at the same time are indicated by the absence of edge between the two vertexes in the graph.



- If there is an edge between the two vertexes, this means that those two courses cannot be scheduled at the same time. I will use Graph coloring for this. By coloring the vertexes, which are edge between each other, differently, I will prevent the exams that can be taken together at the same time. The fact that the colors of the two vertexes are the same will mean that the exam hours of the two courses may be the same. Because these two courses cannot be taken at the same time, there will be no conflict.

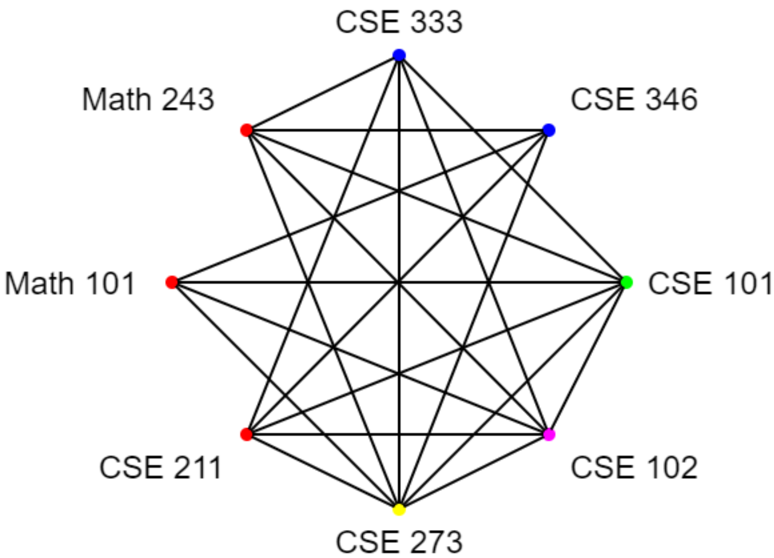


Figure 6: The Hamilton circuit for Problem 2.c

- Red, blue, yellow, green and purple colors represent different time intervals. We solved the exam schedule using 5 different time intervals.

Problem 4: Applications for Hasse Diagram of Relations (40 points)

Remember the Problem 3 in Homework 2.

Write an algorithm to draw Hasse diagram of the given relations in "input.txt" which is given for HW2.

Your code should meet the following requirements, standards and accomplish the given tasks.

- Read the relations from the text file "input.txt". You can use your code from HW2 if you implemented to read the file. If you didn't implement it, please check HW2 to learn how to read the relations from the file.
- Determine each relation in "input.txt" whether it is reflexive, symmetric, anti-symmetric and transitive with your algorithm from HW2.
- In order to draw Hasse diagram, each relation must be POSET. Hence, the relation obeys the following rules:
  - Reflexivity
  - Anti-symmetric
  - Transitivity

  If the relation is not a POSET, your algorithm is responsible to CONVERT it to POSET.
  - If the relation is not reflexive, add new pairs to make the relation reflexive.
  - If the relation is symmetric, remove some pairs which make the relation symmetric. For instance, if the relation has (a, b) and (b, a), remove one of them randomly.
  - If the relation is not transitive, add new pairs which would make the relation transitive.
- After the relation becomes POSET, your algorithm should obtain Hasse diagram of the relation and write the diagram with the following format.
  - An example of the output format is given in "exampleoutput.txt". The file has the result of the first relation in "input.txt".
  - In "output.txt", each new Hasse diagram starts with "n".
  - The relation is (a, a), (a, b), (a, e), (b, b), (b, e), (c, c), (c, d), (d, d), (e, e)
  - The relation is already a POSET so we don't need to add or remove any pairs.
  - After "n", write the POSET in the next line as it is shown in "exampleoutput.txt".
  - Since the relation is POSET, it becomes (a, b), (b, e), (c, d) after removing reflexive and transitive pairs.
  - The following lines give each pair of Hasse diagram.
- You can implement your algorithm in Python, Java, C or C++.
- **Important:** Put comments almost for each line of your code to describe what the line is going to do.
- You should put your source code file (file name is problem1.{$.c$, $.java$, $.py$, $.cpp$}) and output.txt into your homework zip file (check Course Policy).