BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
I SEMESTER 2018-2019
CS F222 - Discrete Structures for Computer Science
Mid Semester Test (Closed book)

Date: 09th October, 2018 **Time: 90 minutes** **MM: 60**

1. Give an inductive proof that the Fibonacci numbers F_n and F_{n+1} are relatively prime or co-prime for all $n \geq 0$. Two numbers $\boldsymbol{a}$ and $\boldsymbol{b}$ are said to be relatively prime if $\gcd(a, b) = 1$, or they have no common factors other than 1. The Fibonacci numbers are defined as follows:

$$F_0 = 0, \quad F_1 = 1 \qquad F_n = F_{n-1} + F_{n-2}$$

[5]

Sol:

- Base Case: The first two consecutive Fibonacci numbers are F0 and F1. So, the base case is n= 0, F0 = 0 and F1 = 1, so, $gcd(F_0, F_1) = gcd(0,1) = 1$. So they are relatively prime (1 Mark)
- Inductive Step: Let $Gcd(F_k, F_{k-1}) = 1$

$$\begin{aligned} Gcd(F_{k+1}, F_k) &= Gcd(F_k, mod(F_{k+1}, Fk)) \\ &= Gcd(F_k, mod(F_{k-1} + F_k, F_k) \\ &= Gcd(F_k, mod(F_{k-1}, F_k) \\ &= Gcd(F_k, F_{k-1}) \\ &= 1 \end{aligned}$$

(4 Marks)

2. **Delhi Metro Network:** There are many different lines (represented by different colors), which connect stations on different parts of the NCR region. A line may intersect with other line(s) at some stations.
   (a) Define a relation R among the stations on the Metro Network. Please note that the relation R you choose to define should be obvious by looking at the Metro Network.
   (b) Using the relation R, as defined in (a), answer the following questions:
      i. Given two stations, figure out whether it is possible to travel between the stations. Give details.
      ii. Given two stations, how will you find the minimum number of hops you need to travel between the stations? A hop is a change of metro train. Give details.
      iii. Is R, as defined in (a), an equivalence relation? Justify.
      iv. Interpret R^n and R^* in context of the problem.

[2+2+2+2+2]

Sol:

Correct Answer: R = { (x,y) | station x and station y are on the same line} (2 marks)

Incorrect answers:
a) R = { (x,y) | There exist a path between station x and station y } (0 marks)
b) R = { (x,y) | x and y are consecutive stations } (0 marks)

(i) Let x and y be the two stations. Calculate the value of M_R, M_{R^2} ………..M_{R^i} iteratively until you get M_{R^i}(x,y) = 1. If we reach to M_{R^n}(x,y) = 0 implies that the two stations are not reachable from each other.

(2 marks)

(ii)

Correct answer: Again, let x and y be the two stations. Let i be the first index s.t M_{R^i}(x,y) = 1, then the i-1 is the minimum number of hops required to travel between the two given stations.

(2 marks)

Incorrect answer

R^*

(0 marks)

(iii) No, it is not transitive

(2 marks)

(iv) R^n={ (x,y) | station x is reachable from station y by changing exactly n-1 number of hops }

R^* = { (x,y) | it is possible to travel from x to y by making as many hops as necessary}

(2 marks)

3. **Create your own Twitter: MyTwitter!** In MyTwitter, anybody can create an account. Once an account is created, a user can start a blog on any topic by creating a new MyHashTag. To make MyHashTag visible to other users, the creator must post a blog under that MyHashTag. A user can post on any existing MyHashTags created by other users. A user U_i becomes a ***follower*** of the user U_j, if U_i posts on any MyHashTag created by U_j. Model the ***follower*** relation of MyTwitter as a suitable type of graph which enables you to answer the following questions:
   (a) Find out all the users being followed by a given user
   (b) Find out all the followers of a given user
   (c) Find out the user who you are following for maximum number of MyHashTags. Also, find the MyHashTag on which the given user has maximum posts.
   (d) Is the ***follower*** relation reflexive, symmetric, anti-symmetric, and transitive?

Ans: We can model the follower relation using a directed, multigraph with self-loops in the following manner.

G = (V,E), where

V = { u | u has an account in *MyTwitter}*

E = { (u_i, v_j) | **u** has posted **i** comments on the j^{th} *MyHashtag* of **v**}

(6 marks for defining the graph correctly as Weighted, Directed and Multigraph)

i. Find out all the users being followed by a given user
   Generate an spanning simple subgraph H(V, E') from G(V, E) s.t V = V' and $(u, v) \in E'$ if there exist a pair (i,j) such the $(u_i, v_j) \in E. Let$ u be vertex corresponding to the given user in graph H, then the N(u) is the set of users followed by u, where N(u) = { v | uv *is an edge in* H}

(1 mark)

ii. Find out all the followers of a given user
   Generate an spanning simple subgraph H(V, E') from G(V, E) s.t V = V' and $(u, v) \in E'$ if there exist a pair (i,j) such the $(u_i, v_j) \in E. Let$ u be vertex corresponding to the given user in graph H, then the M(u) is the set of all followers of u, where M(u) = { v | vu *is an edge in* H}

(1 mark)

iii. Find out the user who you are following for maximum number of MyHastags. Also, find the MyHashtag on which the given user has maximum posts.

user who you are following for maximum number of MyHastags

1. Let u be the vertex corresponding to the user
2. Define **OUT_DEGREE(u,v)** as the number of edges from u and v in G
3. *MAX = 0, ANSWER* = Φ
3. **For** each vertex $v \in V$ $\boldsymbol{do}$
4. If OUT_DEGREE(u, v) > MAX
5. MAX = OUT_DEGREE(u, v), *ANSWER* = v
7. **EndFor**
8. Return *ANSWER;*

MyHashtag on which the given user has maximum posts

1. Let u be the given user. Type equation here.
2. MAX = 0, MAX_MYHASHTAG
2. For each MyHashtag $\boldsymbol{j}$ of user $\boldsymbol{u}$ **do**
3. Find the set of users S = $\{\ v \mid (v_i, u_j) \in G\ \}$
4. Let Index(v) = i, s.t $(v_i, u_j) \in G$
4. For each element v in S
5. If Index(v) > MAX, MAX = Index(v), MAX_MYHASHTAG = j

6. **EndFor**
7. Return MAX_MYHASHTAG

(1 mark)

iv. Is the **follower** relation reflexive, symmetric, anti-symmetric, and transitive?
Reflexive: No, what if a user has not created any MyHashTags
Symmetric: No, A following B does not imply that B also follows A
Anti-Symmetric: No, A following B does restrict B to follow A
Transitive: No, A following B and B following C, does not imply that A follows C

(1 mark)

** No marks have been given for directly writing the properties instead of using the graph modeled by you.

4. An unusual species inhibits the forest surrounding a Functional City. Each member of the species can take one of three possible forms, called *Schemander, Haskeleon and Camlizard.* In January of every year each individual undergoes an "evaluation" – a process by which the individual splits into two individuals, whose form depend upon the form of the individual.
   - A *Schemander* splits into a *Schemander* and *Haskeleon*
   - A *Haskeleon* splits into a *Schemander* and *Camlizard*
   - A *Camlizard* splits into a *Schemander* and *Haskeleon*

   It is known that in June of year 0 (zero), the population consisted of a single *Schemendar*. Assume that no individual ever dies and all individuals successfully undergo evaluation exactly once every January.

   (a) Let S_n, H_n, and C_n be the number of *Schemanders, Haskeleons, and Camlizards*, respectively, in June of year $\boldsymbol{n}$. Express S_n, H_n, and C_n in terms of $S_{n-1},\ H_{n-1},\ and\ C_{n-1}$ for $n > 0$.

   $S_n = S_{n-1} + H_{n-1} + C_{n-1}$
   $H_n = S_{n-1} + C_{n-1}$
   $C_n = H_{n-1}$

(5 marks)

*** No Partial Marking

(b) Let $T_n = S_n + H_n + C_n$ be the total number of individuals in June of year n. Guess the closed form equation for T_n (no need to derive) and use induction to prove your answer for all $n \geq 0$.

We can see that the population doubles every year so, and $T_n = 2^n$
Proof

Basis: $T_0 = 1$
Inductive Step:
$T_n = S_n + H_n + C_n$
$T_n = (S_{n-1} + H_{n-1} + C_{n-1}) + (S_{n-1} + C_{n-1}) + (H_{n-1})$
$T_n = 2(S_{n-1} + H_{n-1} + C_{n-1})$
$T_n = 2(T_{n-1})$

(3 marks)

*** 1.5 marks if the answer is correct but did not provide the closed form formula

(c) Show that $H_n = T_{n-1} - H_{n-1}$ for $n > 0$.
$H_n = S_{n-1} + C_{n-1}$
$H_n = (S_{n-1} + H_{n-1} + C_{n-1}) - H_{n-1}$
$H_n = T_{n-1} - H_{n-1}$

(2 marks)

*** No Partial Marking

5. (a) Use Warshall's Algorithm to find the transitive closure of relation R, represented by the matrix given below. Show all the intermediate matrices, (W_i), that you get in the process.

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

Sol:

$$W_0 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}, \quad W_1 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix},$$

$$W_2 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \quad W_3 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

(b) Warshall's algorithm, in some sense, determines the existence of paths between vertices of a graph. Suppose, we keep track of the lengths of paths also, then we can get the length of the shortest path between all vertices in a graph. Modify the Warshall's algorithm to get the shortest distance between two vertices.

Sol:

- First, instead of initializing W to be M_R, we initialize it to be M_R with each 0 replaced by ∞ and each 1 by the weight of the corresponding edge.
- Second, the computational step becomes $w_{ij} := min(w_{ij}, w_{ik} + w_{kj})$.

(c) Use the modified Warshall's Algorithm to calculate the shortest distance between every two vertex of the graph shown below:

[5+3+2]

6. In the CS department of BITS-Pilani, n professors $(P_1, P_2, \ldots . P_n)$ are available to teach n subjects $(S_1, S_2 \ldots S_n)$ offered by the department in a semester. Each professor can teach only a limited set of courses, depending upon his/her expertise. The list of course that each professor can teach is available with the department. The HoD is interested to know whether it is possible to assign the courses to the professors (only one course per professor per semester) such that all the courses can be offered in the coming semester. How will you model this problem using graphs? Also, propose a solution.

[10]

1. Model the problem as bipartite graph with V1 as the set of professors and V2 as the set of courses. There exist an directed edge between a node u of type V1 to the node v of type V2 if professor u can teach course v.

(**2 marks**)

2. The underlying problem in the question is to find out whether a perfact matching exist in the bipartite graph as defined above.

(**1 mark**)

3. Let G be a finite bipartite graph with bipartite sets V1 and V2 (G:= (V1 + V2, E)). For a set W of vertices in V1, let N_G(W) denote the neighborhood of W in G, i.e. the set of all vertices in V2 adjacent to some element of W then there is a matching that entirely covers V1 if and only if for every subset W of V1:

| W | ≤ | N_G(W) |

In other words: every subset W of V1 has sufficiently many adjacent vertices in V2. The above condition is a necessary and sufficient for the existence of a perfect matching in the graph.

(4 marks)

4. Provide an algorithm to test whether the above condition is satisfied in a given graph.

(3 marks)

Incorrect answer

Why the greedy algorithm does not work?

Consider the following greedy algorithm

1)Select a professor which can teach minimum number of courses

2)Greedily allocate a course to that professor (from the ones which he/she can teach)

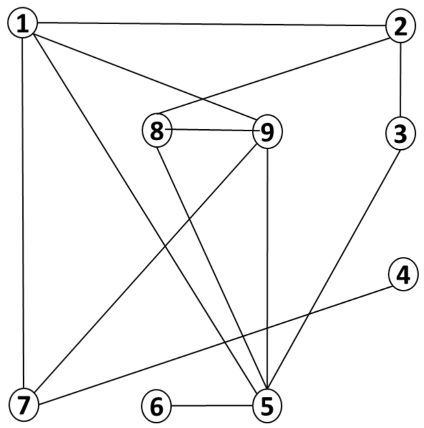
3)Remove both the professor and the course from the graph

4) Repeat step 1,2,3

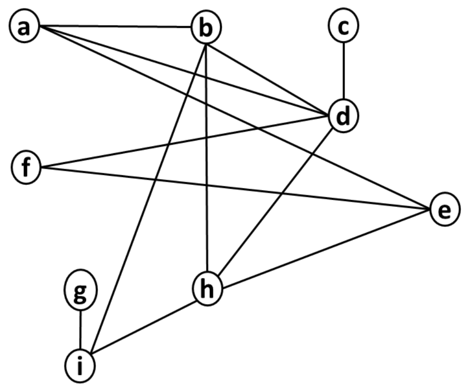
The above greedy approach will not work on bipartite matching.

Here is an example - nodes on the left are P1, P2, P3 and P4 and on the right are x, y, z, t. Connect each of A, B, C with each of x, y, z (so 9 edges here) then connect D with t and A with t. As a result, you have 3 nodes on the right with in- degree 3(x, y, z) and one with in-degree 2(t). So you choose t and you choose one node on the left at random - this may be A or D. Problem is that if you select A, your max matching will be of size 3, while the real answer is 4(by selecting D).

7. Determine whether the graphs, **G** and **H**, are isomorphic. If yes, provide an isomorphic mapping. Otherwise, give a valid argument to show that they are not isomorphic.



G



H

Sol:
1-h, 2-e, 3-f, 4-g, 5-d, 6-c, 7-i, 8-a, 9-b

[5marks]

** No partial marking

*****