Jennifer MacDonald
604501712
03/08/17

Homework 3

CS 249: Big Data Analytic, Winter 2017
Instructor: Prof. Wei Wang

Due on: Wednesday, March 8, 2017

Instructions

- Please be short and concise in your answers. Do not present unnecessary details.
- Please submit your submission online in CCLE before the due date
- Please mention your full name, UCLA ID and your discussion section on the first page.

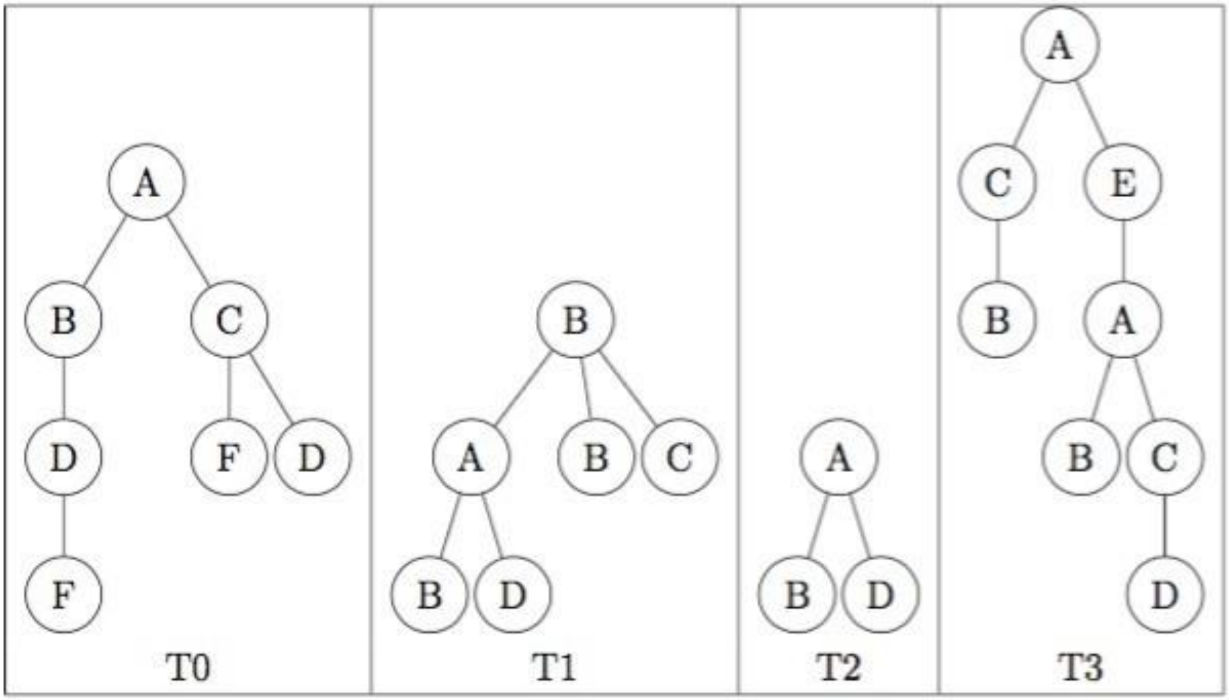


Figure 1: Tree for Problem A

Problem A. Tree Mining (45 pts)

1. Using DFS traversal, show the string representation of each tree. (8 pts)

T0: A B D F -1 -1 -1 C F -1 D -1 -1
T1: B A B -1 D -1 -1 B -1 C -1
T2: A B -1 D -1
T3: A C B -1 -1 E A B -1 C D -1 -1 -1 -1

2. Is T2 an induced subtree of T0? (4 pts)

No

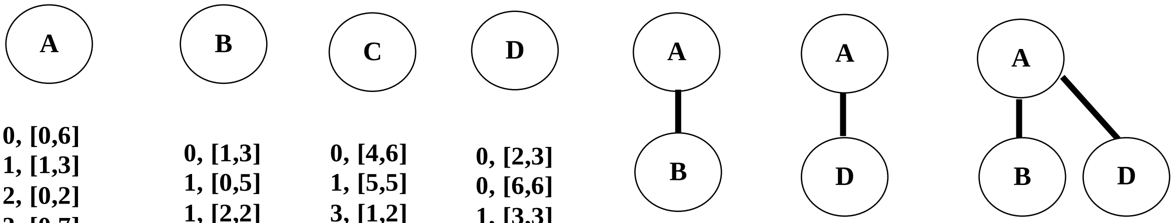
3. Is T2 an embedded subtree of T0? (4 pts)

Yes

4. Assume min supp = 3 for the following problems. Show the scope of each node for all four trees. (10 pts)

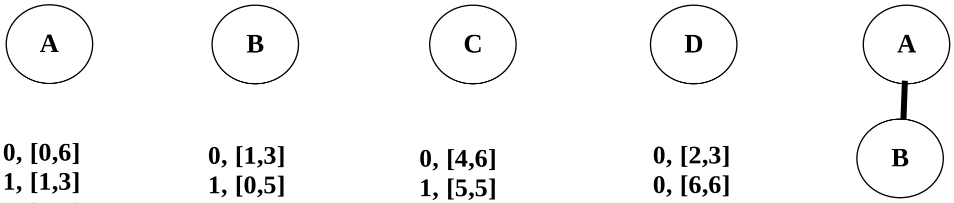
0, [0,6]	**0, [1,3]**	**0, [4,6]**	**0, [2,3]**	**3, [3,7]**	**0, [3,3]**
1, [1,3]	**1, [0,5]**	**1, [5,5]**	**0, [6,6]**		**0, [5,5]**
2, [0,2]	**1, [2,2]**	**3, [1,2]**	**1, [3,3]**		
3, [0,7]	**1, [4,4]**	**3, [6,7]**	**2, [2,2]**		
3, [4,7]	**2, [1,1]**		**3, [7,7]**		
	3, [2,2]				
	3, [5,5]				

5. Show the frequent embedded subtrees and their corresponding scope-lists. (10 pts)



0, [0,6]	**0, [1,3]**	**0, [4,6]**	**0, [2,3]**	**0, 0, [1,3]**	**0, 0, [2,3]**	**0, 01, [6,6]**
1, [1,3]	**1, [0,5]**	**1, [5,5]**	**0, [6,6]**	**1, 1, [2,2]**	**0, 0, [6,6]**	**1, 12, [3,3]**
2, [0,2]	**1, [2,2]**	**3, [1,2]**	**1, [3,3]**	**2, 0, [1,1]**	**1, 1, [3,3]**	**2, 01, [2,2]**
3, [0,7]	**1, [4,4]**	**3, [6,7]**	**2, [2,2]**	**3, 0, [2,2]**	**2, 0, [2,2]**	**3, 02, [7,7]**
3, [4,7]	**2, [1,1]**		**3, [7,7]**	**3, 0, [5,5]**	**3, 0, [7,7]**	**3, 05, [7,7]**
	3, [2,2]			**3, 4, [5,5]**	**3, 4, [7,7]**	**3, 45, [7,7]**
	3, [5,5]					

6. What are the frequent induced subtrees? (7 pts)



0, [0,6]
1, [1,3]
2, [0,2]
3, [0,7]
3, [4,7]

0, [1,3]
1, [0,5]
1, [2,2]
1, [4,4]
2, [1,1]
3, [2,2]
3, [5,5]

0, [4,6]
1, [5,5]
3, [1,2]
3, [6,7]

0, [2,3]
0, [6,6]
1, [3,3]
2, [2,2]
3, [7,7]

0, 0, [1,3]
1, 1, [2,2]
2, 0, [1,1]
3, 0, [2,2]
3, 0, [5,5]
3, 4, [5,5]

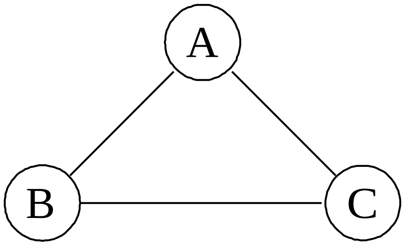


Figure 2: Graph for Problem B.1

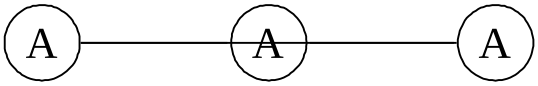


Figure 3: Graph for Problem B.2

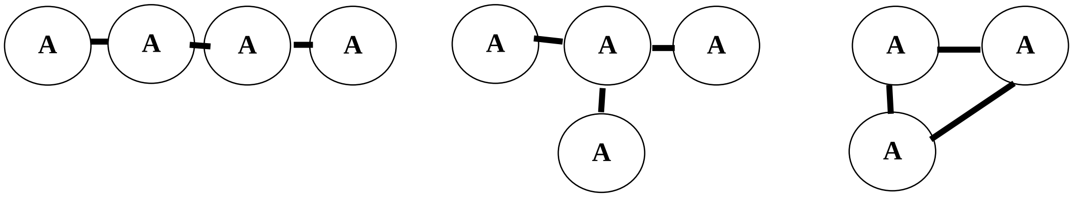
Problem B. Graph Mining (25 pts)

1. Consider the graph in Figure 2. If all the edges have the value X and assuming $0 < A < B < C < X$, what is the CAM of the graph? (10 pts)

CXAXXB
CXBXXA
B > A

C		
X	**B**	
X	**X**	**A**

2. Consider the graph in Figure 3. If all the edges have the value X, how many candidates are generated by self-join using the FSG algorithm? (10 pts)



There are 3 candidates generated by self-join using the FSG algorithm.

3. Name and briefly describe all three cases for the FSG algorithm. (5 pts)

Case 1: identical vertex labels. This is when you join a graph with itself.

Case 2: core contains identical labels. This is where the graphs have a common core, but the node connected to the common core is different.

Case 3: core multiplicity. The graphs share a common core but have nodes attached to different nodes in the common core.

Problem C. Programming Assignment (70 pts)

Implement a program that solves a general case of CAM.

1. Implement a function named convertToMatrix that takes in a array of edges and converts it into a CAM adjacency matrix. Each edge will be represented as a 3-item list. ie: [0,1,'X'] will represent an edge from 0 to 1 with a value of X. ['B','C',2] will represent an edge from 'B' to 'C' with a value of 2. You may assume the graph represented in an undirected graph and each node is unique in the graph. The output of your program must be 2D array or list in your language. In addition, the function must also print the code equivalence of the adjacency matrix in string format. (Characters are not case-sensitive)

2. Implement a function called generateCAM that takes in two CAM adjacency matrix representation you wrote in C1 that creates a new matrix using CAM to handle join case 1 (Both A and B have at least two edge entries in the last row) and join case 2 (A has at least two edge entries in last row but B has only one).

   The output of your program must be 2D array or list in your language. If the join case is not 1 or 2, output a blank 2D array or list.

   The function must also print a string of the join case it matched. If the graph matches join case 1, print "Case 1". If the graph matches join case 2, print "Case 2". If the graph matches join case 3a, print "Case 3a". If the graph matches join case 3b, print "Case 3b". Note you do not have to implement the actual functionality of case 3 join.

You may program in either R, MATLAB, OCTAVE, or python. You may only use the core library functions provided by the language. If you have any doubts of what functions you can use please contact the TA. Any ambiguity should be stated in your README file.