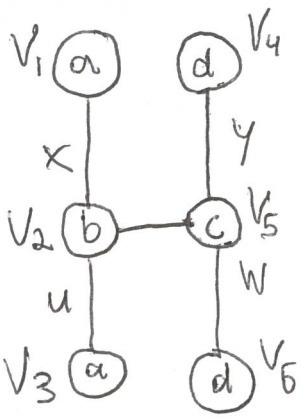
1. Consider the following graph:



Using the <u>vertex invariants scheme</u>, construct the (minimal) canonical label for this graph. Show Your Work in the application of this method; otherwise you will receive no credit for this problem. (6.5 points)

<u>Hint:</u> You will need to consider different permutations of the partitioned matrix!

<u>Solution:</u>

1	x	0	0	0	0	0
2	x	0	u	0	z	0
3	0	u	0	0	0	0
4	0	0	0	0	y	0
5	0	z	0	y	0	w
6	0	0	0	0	w	0

degree 1: $[V_1, V_3, V_4, V_6]$

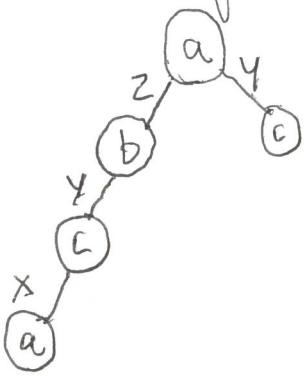
degree 3: $[V_2, V_5]$

See Excel sheet for matrices.

* Idw not sure if swapping starting w/ V_6 and V_4 instead of V_1 and V_3 was needed since it could quickly eliminate them based off of their matrices starting w/ a 4th zero but it didn't see a downside to including them.

∴ The minimal canonical label is 000u000x000w0yz

2. Consider the following graph:



Using the minimum depth-first search code (M-DFSC) strategy, find the minimal canonical label for this graph. Show every possible M-DFSC encoding for this graph so that we can see that your choice of a canonical label is the minimal M-DFSC (6.5p

Solution:

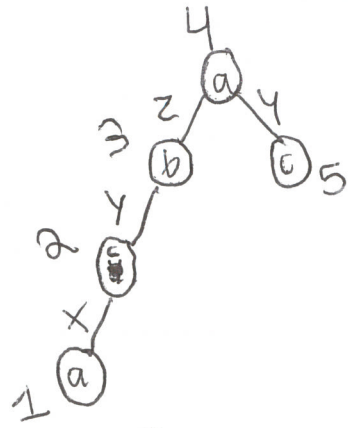
Excel sheet for problem 2 shows:

① (1,2,a,x,c) (2,3,c,y,b) (3,4,b,z,a) (4,5,a,y,c
② (1,2,c,x,a) (1,3,c,y,b) (3,4,b,z,a) (4,5,a,y,c
③ (1,2,c,y,b) (2,3,b,z,a) (3,4,a,y,c) (1,5,c,x,a
④ (1,2,b,y,c) (2,3,c,x,a) (1,4,b,z,a) (4,5,a,y,c
⑤ (1,2,a,z,b) (2,3,b,y,c) (3,4,c,x,a) (1,5,a,y,c
⑥ (1,2,b,z,a) (2,3,a,y,c) (1,4,b,y,c) (4,5,c,x,a
⑦ (1,2,a,y,c) (1,3,a,z,b) (3,4,b,y,c) (4,5,c,x,a
⑧ (1,2,c,y,a) (2,3,a,z,b) (3,4,b,y,c) (4,5,c,x,a

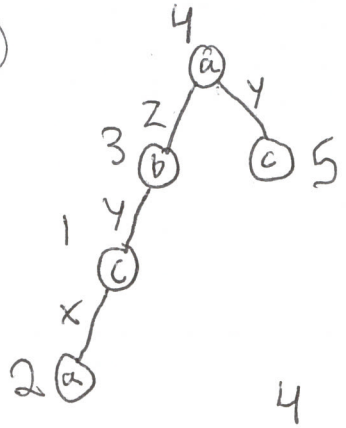
See page 2 for Vertex input possibilities.

∴ minimal canonical label is (1,2,a,x,c)

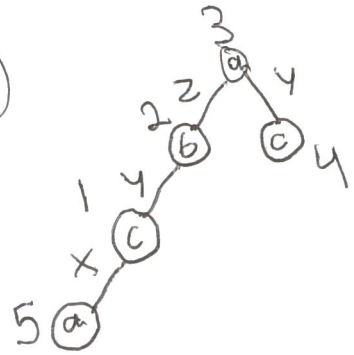
1


4
a
3
2
b
y
c
5
2
y
s
x
1
a

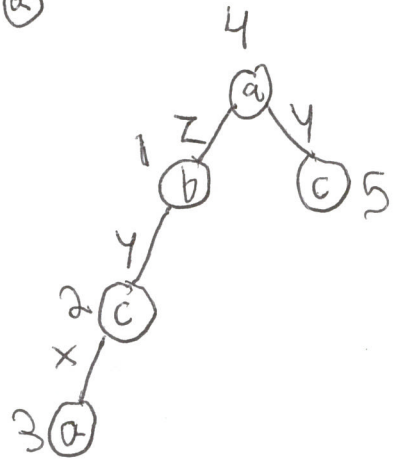
2


4
a
3
2
b
y
c
5
1
y
c
x
2
a

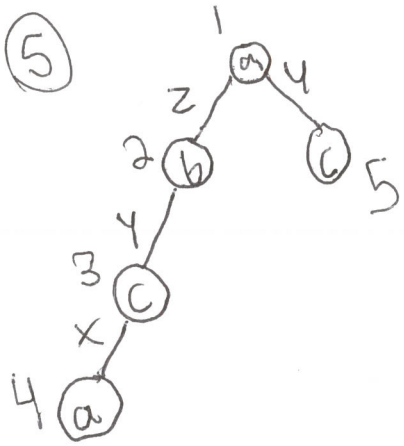
3


3
a
2
2
b
y
c
4
1
y
c
x
5
a

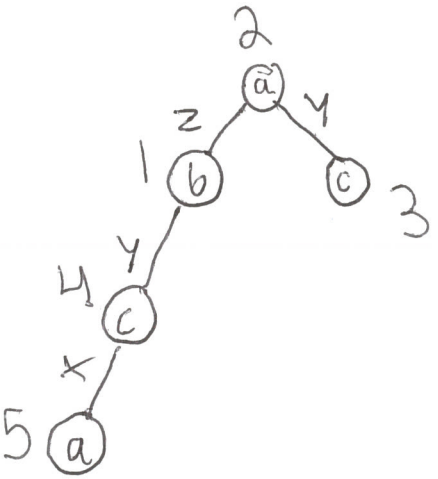
4


4
a
1
2
b
y
c
5
y
2
c
x
3
a

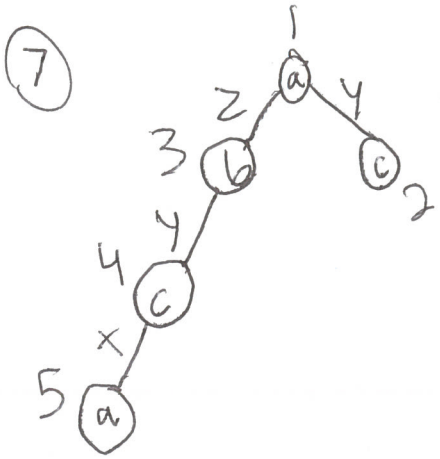
5


1
a
2
2
b
y
c
5
3
y
c
x
4
a

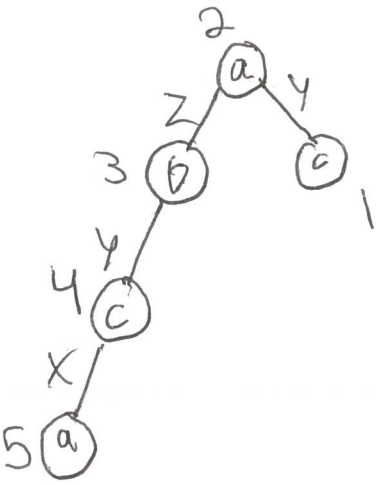
6


2
a
2
1
b
y
c
3
4
y
c
x
5
a

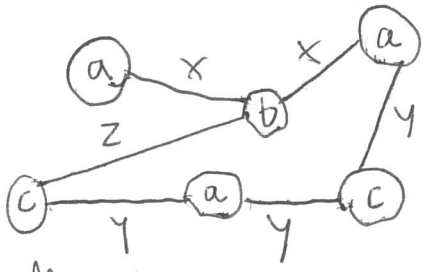
7


1
a
2
3
b
y
c
2
4
y
c
x
5
a

8

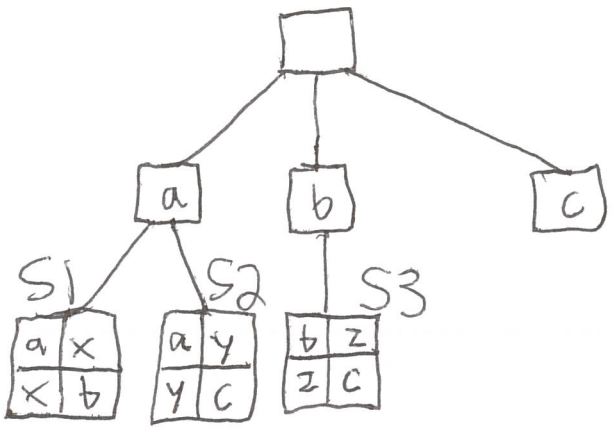

2
a
2
3
b
y
c
1
4
y
c
x
5
a

3. Consider the following graph: 1/5



Shown below is the start of (what is intended to be) a DAG of canonical adjacency matrices for connected subgraphs of this graph using extension and join operations, construct only one more level in the tree off of S1, S2, and S3 (i.e., only create child nodes for the nodes labeled S1, S2, and S3); you are Not being asked to construct the entire DAG for the graph. Clearly explain the join and extension operations you are doing for each node. You must give the canonical adjacency matrix (as calculated including diagonal entries) AND draw each subgraph node in your solution. (4.5 pts)

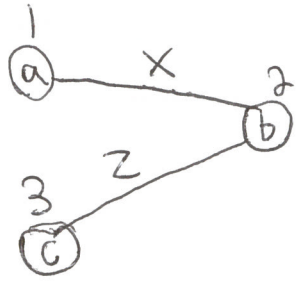
When you're finished, re-draw the tree showing the children of S1, S2, and S3 w/ their matrices and pictures of their graphs!



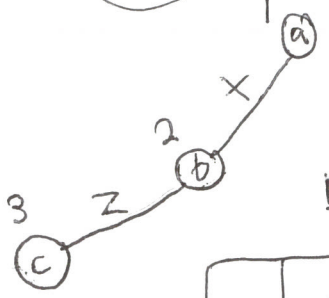
3. Solution:

S_1

a	x
x	b

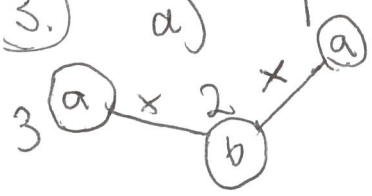
① 

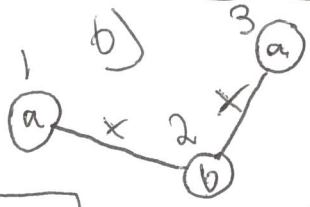
a	x	0
x	b	z
0	z	c

② 

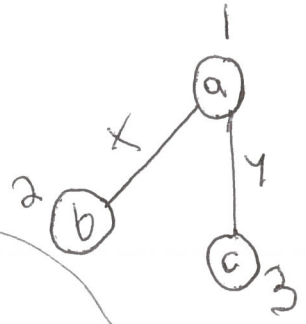
redunda

a	x	0
x	b	z
0	z	c

3. a) 

b) 

a	x	0
x	b	x
0	x	a

④ 

a	x	y
x	b	0
y	0	c

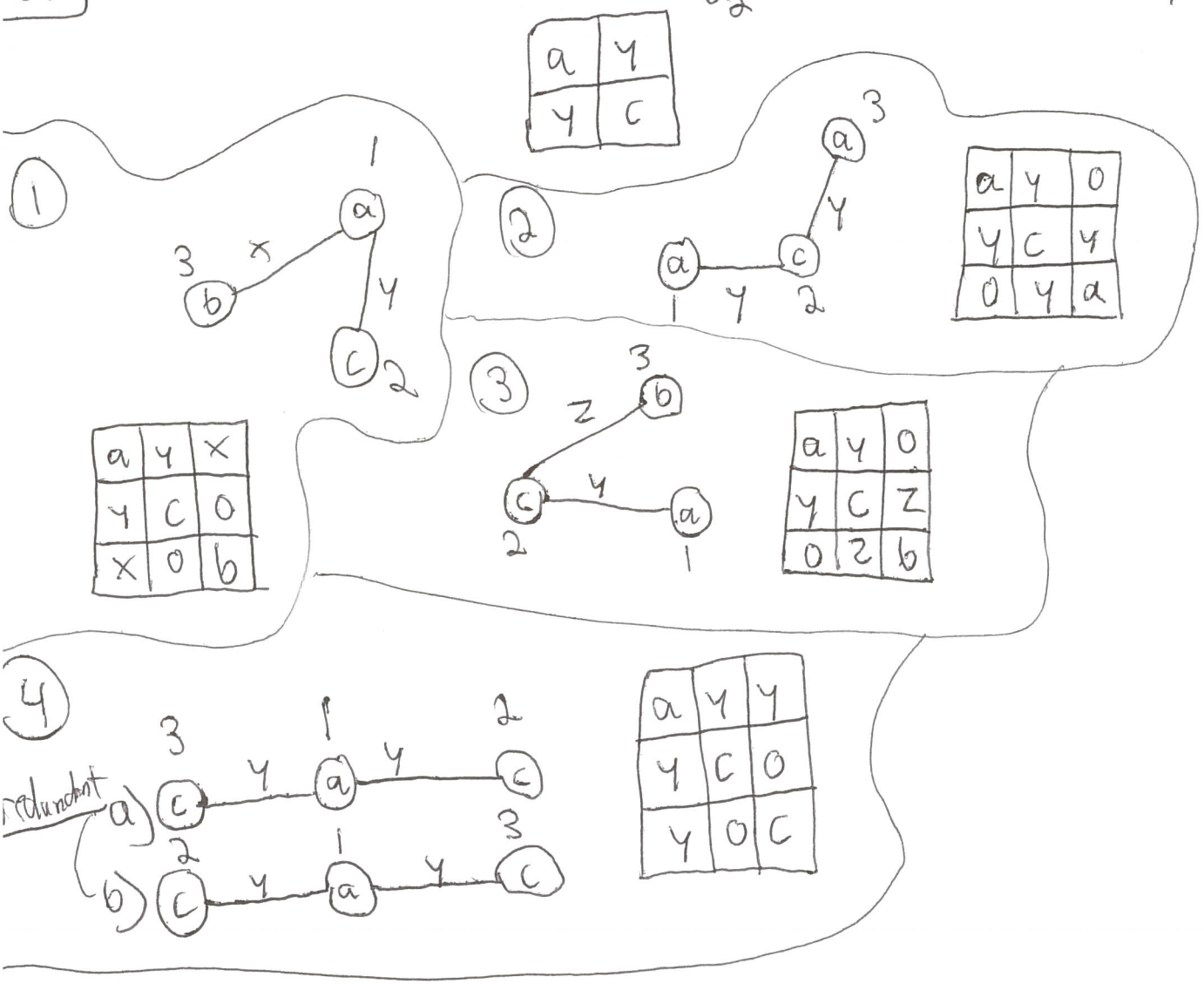
① extension because vertex 3 added from 2 along w/ its edge

② extension because vertex 3 added from 2 along w/ its edge

③ extension because vertex 3 added from 2 along w/ its edge

④ join because another edge, edge c, is added, Vertex 3 and associated edge 3 don't extend off of 2, therefore joined with original subgraph from 1.

3. 3/5



1) join b/c edge b is added off of 1, not 2

2) extension b/c vertex 3 added from 2 along w/ its edge

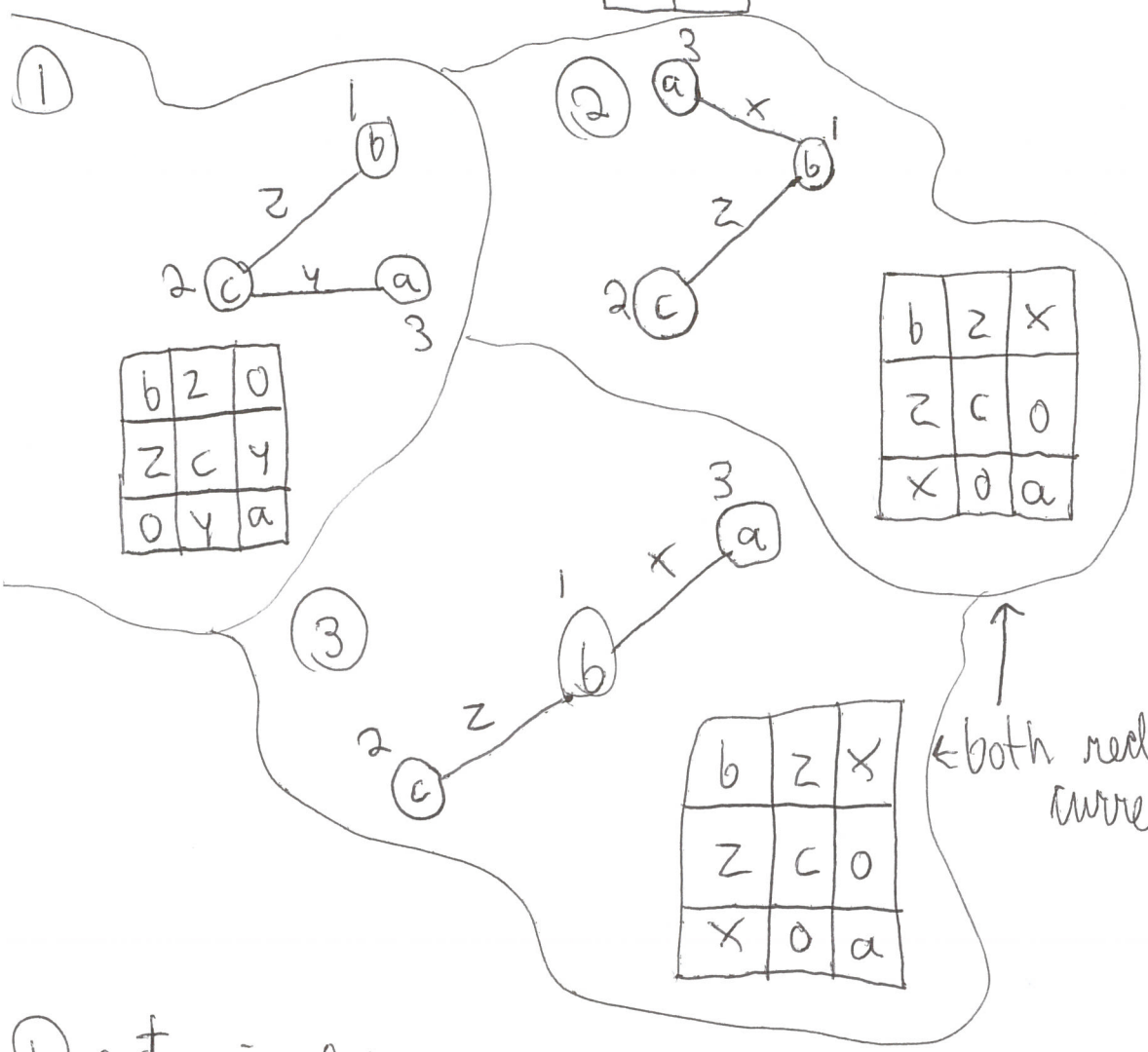
3) extension b/c vertex 3 added from 2 along w/ its edge

4) join b/c edge c is added off of 1, not 2

3.

S_3

b	z
z	c



b	z	0
z	c	y
0	y	a

b	z	x
z	c	0
x	0	a

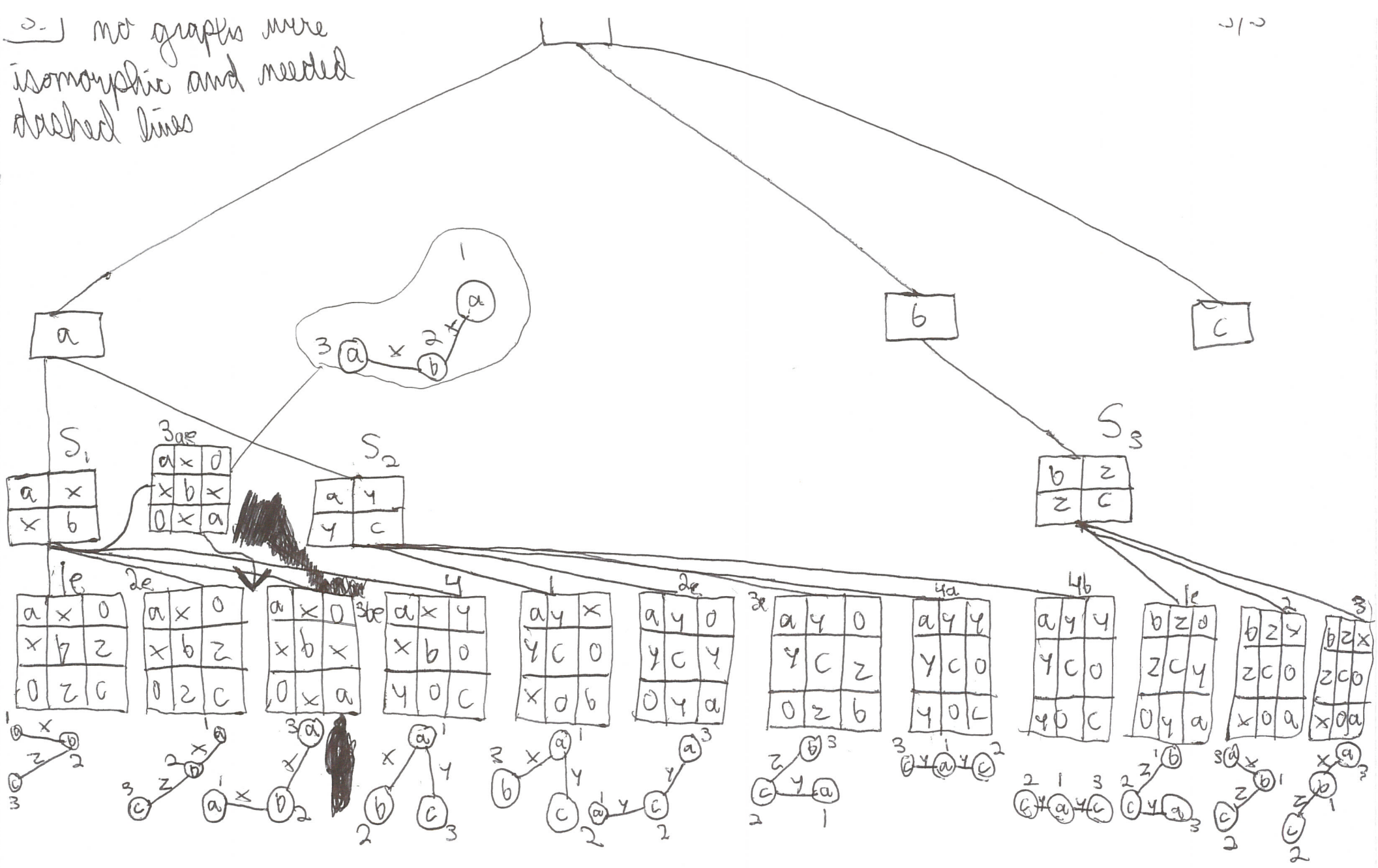
b	z	x
z	c	0
x	0	a

← both redundant at current level

① extension b/c vertex 3 added from 2 along w/ its edge

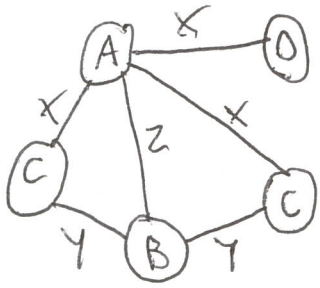
②③ join cases b/c edges (a in both cases) added off of 1, not 2

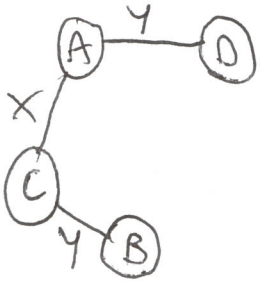
3.) no graphs were isomorphic and needed dashed lines

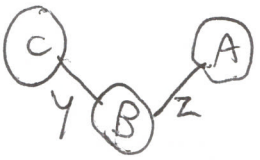


* 1e and 2e redundant, 4a and 4b redundant; 2 and 3 for S_3 redundant - wasn't sure whether to keep both subgraphs in or not decided to because they'd have different children.

4. Consider the following collection of 3 graphs:







Show how gSpan would find all frequent subgraphs using a minimum support threshold of 2. At the conclusion of your work, Draw all of the frequent subgraphs you found. Show all your work in doing the algorithm.

Solution:

Edge	Support Count
A x D	1 X
D x A	1 X
A x C	2 ✓
C x A	2 ✓
C y B	3 ✓
B y C	3 ✓
B z A	2 ✓
A z B	2 ✓
D y A	1 X
A y D	1 X

"0,1" added to beginning to denote the edges starting and ending points.

Winners

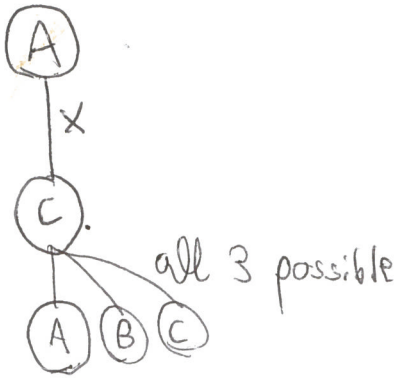
Edge	Support Count
A x C	2
C x A	2
C y B	3
B y C	3
B z A	2
A z B	2

Sorted in lexicographic order →

Edge	Support Count
(0,1,A,x,C)	2
(0,1,A,z,B)	2
(0,1,B,y,C)	3
(0,1,B,z,A)	2
(0,1,C,x,A)	2
(0,1,C,y,B)	3

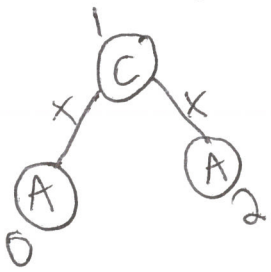
4.

Solution: Start w/ min.-labeled edge in the list, (0, 1, A, x, C)
deepest vertex: C — can't link back to A, can't link back to next vertex [redacted], itself.



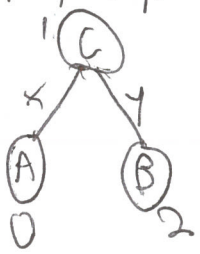
Option 1

(0, 1, A, x, C)
(1, 2, C, x, A)



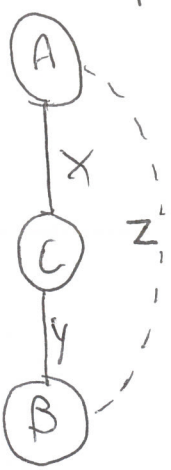
Option 2

(0, 1, A, x, C)
(1, 2, C, y, B)



Option 1: doesn't [redacted] occurs in any of the graphs; doesn't meet minimum support threshold

Option 2: occurs in 2 of the 3 graphs; meets minimum suppo



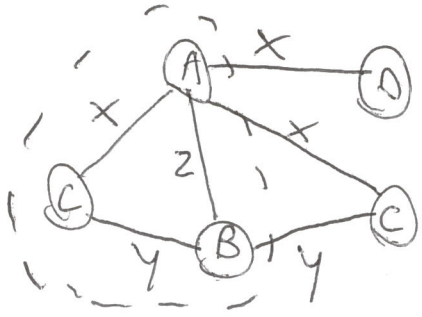
Do to deepest Vertex "B" and try to add an edge back to the "first" vertex (A).

4.

Solution:

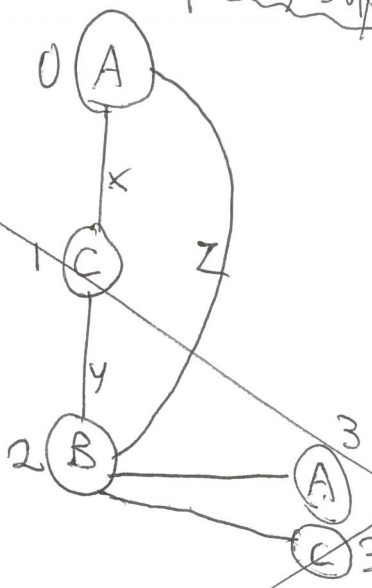
(A, z, B) is a frequent edge. The edge would be denoted (2, 0, B, z, A).

This graph structure can be seen in 1 only.



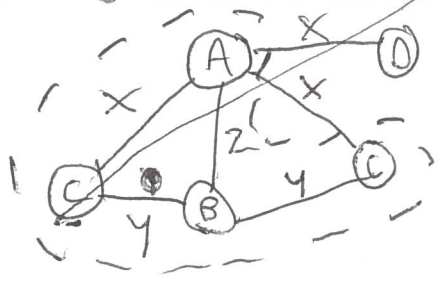
↑ even though graph 3 contains the pattern going from C y B to B z A the whole structure is needed.

What follows isn't frequent, supported



Options: (0, 1, A, z, B)
(0, 1, C, y, B)
(0, 1, B, y, C)
(0, 1, B, z, A)

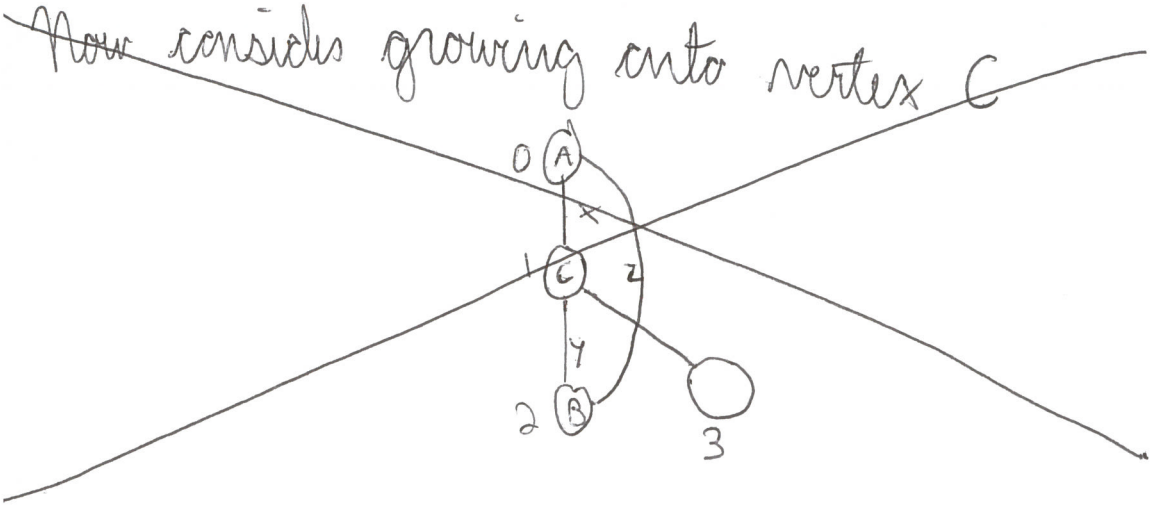
adding (2, 3, B, y, C) only occurs in 1 graph



4. 4/4

Solution:

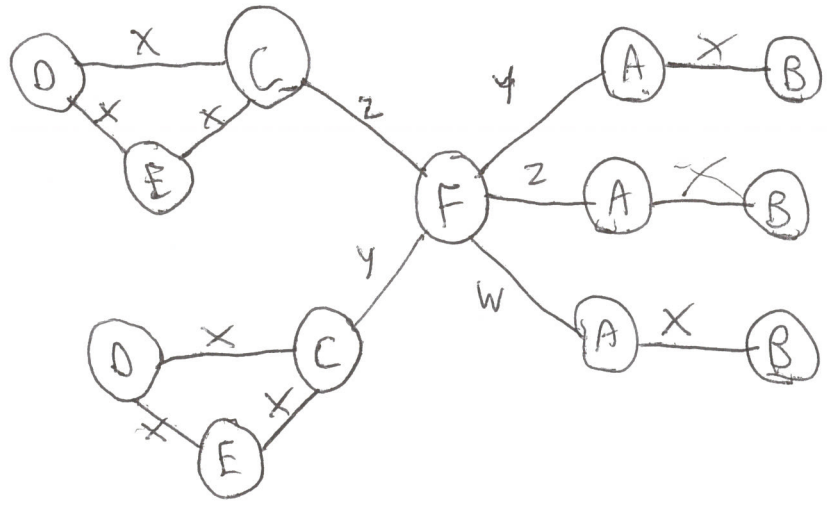
Now consider growing onto vertex C



∴ With a support threshold of 2, the frequent subgraph expands to

5. In class we discussed two different methods for determining which subgraph to choose for compression in the SUBDUE algorithm: Minimum Description Length (MDL) and Size. In the graph given below, use the Size method to determine whether the subgraph containing vertices {A, B} or the subgraph containing {C, D, E} would be chosen for the very first compression of the graph G. Show your work in how the determination is made and show each of the two possible compressed Graphs! (4.5 pts)

Graph G



Note: the formula for the Size method is:

value(S, G) = VE_size(G) / (VE_size(S) + VE_size(G|S))

where VE_size(G) = #vertices(G) + #edges(G)

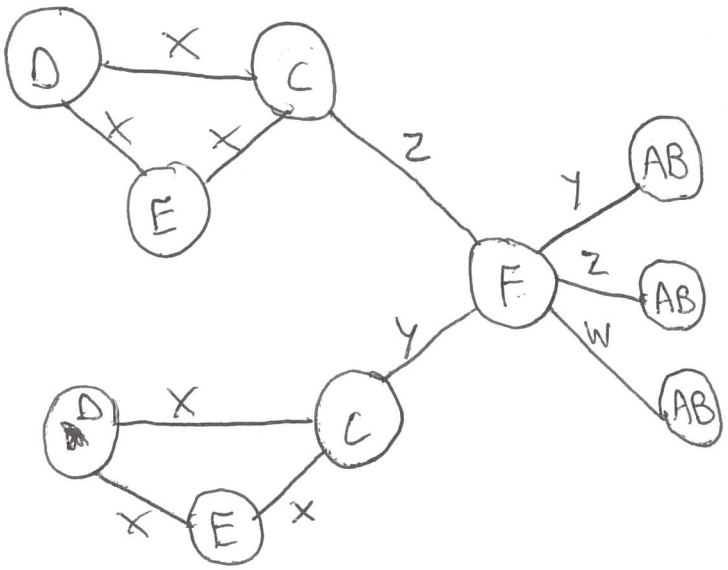
Solution: See next page

5. Solution:

A (occurs 3x)
B (occurs 3x)
C (occurs 2x)
D (occurs 2x)
E (occurs 2x)
F (occurs 1x)

For subgraph containing vertices {A, B}:

would be the best substructure to be added to the queue.

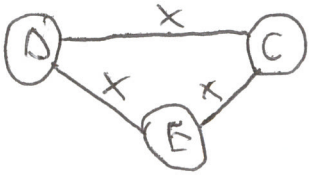


VE_size(D) = 14 + 13 = 27
VE_size(S) = [redacted] 1 + 2 = 3
VE_size(GIS) = 11 + 10 = 21
value(D, D) = 27 / (3 + 21) = 27/24 = 1.125

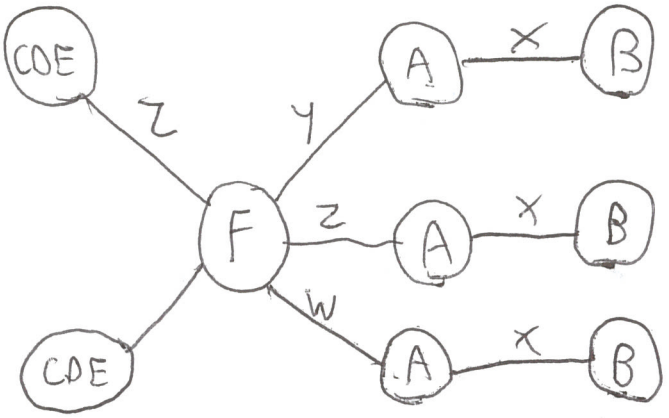
5. 3/3

Solution:

For subgraph containing vertices $\{C, D, E\}$:



would be the best substructure to be added to the queue.



VE_size($\mathcal{D}$) = 27

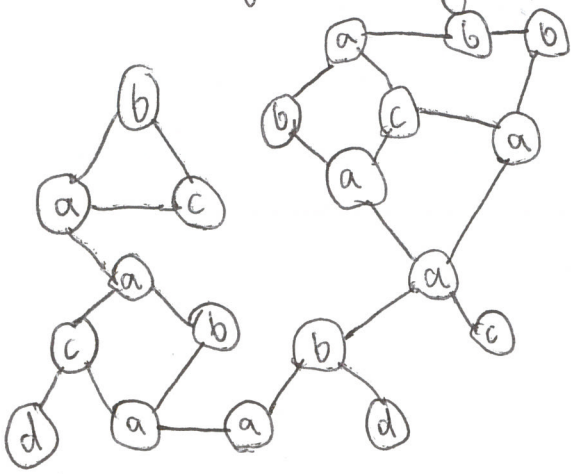
VE_size($\mathcal{S}$) = 3+3 = 6

VE_size($\mathcal{D}|\mathcal{S}$) = 8+9 = 17

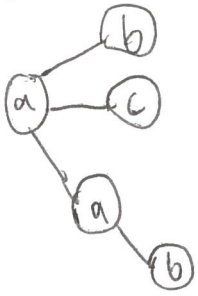
value($\mathcal{D}, \mathcal{S}$) = 27/(6+17) = 27/23 = 1.1739

vertices $\{A, B\}$ have smaller value than $\{C, D, E\}$ so $\{A, B\}$ would be chosen for the first compression of the graph.

6. Consider the following graph: 1/2



Suppose that HSIGRAM is considering the frequency of the subgraph shown below:



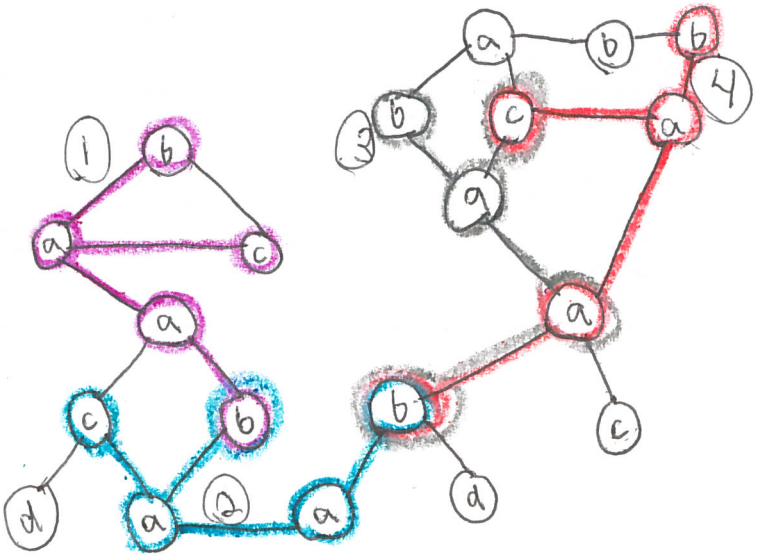
Construct an embedding overlap graph for this subgraph. Then compute the size of the overlap graph. Show how you arrived at your answer (i.e., show the computation of the MIS for each embedding that must be considered in the overlap graph). Use a different color to draw each embedding. [7.5 pts)

Solution:

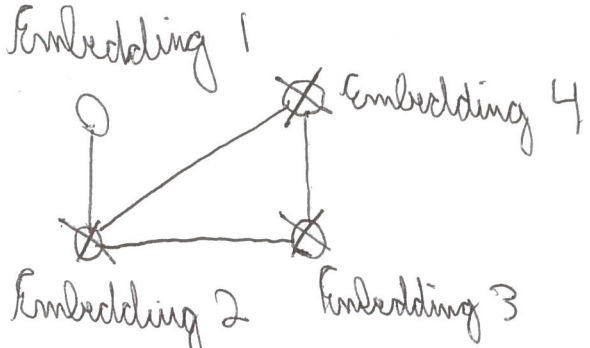
6.

2/2

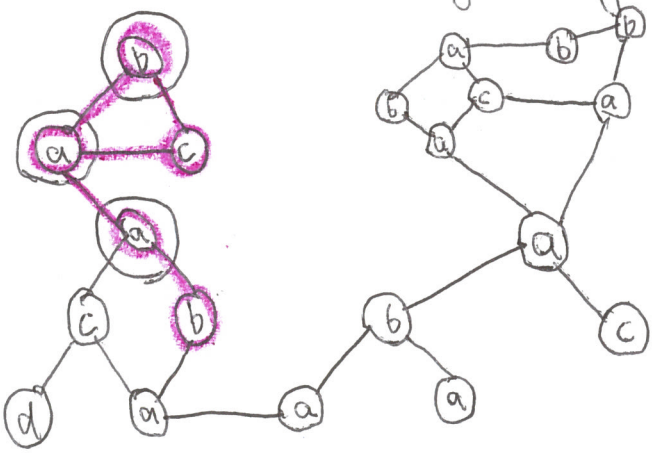
Solution:



Overlap



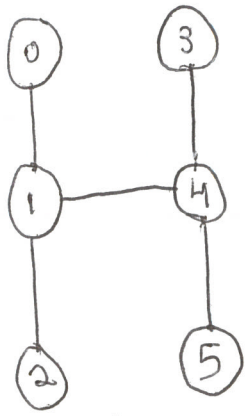
Don't want nodes in the embedding overlap graph w/ degree ≥ 2 since we only want to count edge-disjoint subgraphs.



MCS considering only the vertices in this embedding and all the edges in the graph is size 3.

Since embedding for this graph only includes embeddi
the embedding for this graph is 3

7.] Consider the graph shown below: 1/3



a.) What would be the equitable refinement of the unit partition, $R(\mu)$ according to the Nauty algorithm? Clearly explain how you got your answer (1 pt.)

b) The Nauty algorithm generates a "search tree" w/ a root that is the equitable refinement of the unit partition, $R(\mu)$ (i.e., your answer to part a). Show what the leftmost child (partition) would be off of the root node of that tree. Clearly explain how you got your answer! (4 pts)

Note: You do Not have to generate the entire search tree; this question is only asking for one node of the tree!

c.) In class we discussed that when you execute the autgrp function in Nauty (for example in Python), one of your outputs you get is what is called generators. Give one set that would qualify as a single generator set for the graph given in this problem. (1 pt.)

Solutions: See next page

7.

Solution:

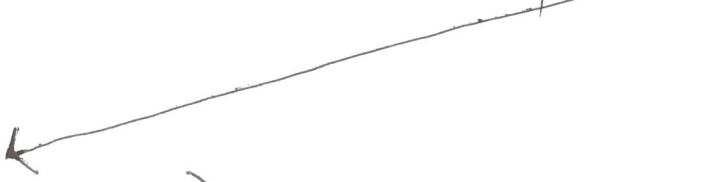
Vertices 0, 2, 3, 5 all have degree 1
Vertices 1 and 4 have degree 3
The unit partition is (0 1 2 3 4 5)

$R(\mu) = (0\ 2\ 3\ 5 \mid 1\ 4)$

- because of their degree groupings

b)

$R(\mu) = (0\ 2\ 3\ 5 \mid 1\ 4)$



(02|35|4|1)

Start from root (0 2 3 5 | 1 4)
start w/ 0, see if it shatters 14
0 has 1 neighbor in 1 and 0 neighbors in 4
List 4 before 1 because 0 neighbors comes before 1 neighbor (want increasing order)
2 has 1 neighbor in 1 and 0 neighbors in 4; same as 0 so put w/ 0
3 has 0 neighbors in 1 and 1 neighbor in 4; different than 0 and 2

7.

3/3

Solution:

5 has 0 neighbors to 1 and 1 neighbor to 4. Put together w/ 3

Does 02 shatter 35, 4, or 1? NO

0 has 0 neighbors w/ 3 and 0 neighbors w/ 5

2 has 0 neighbors w/ 3 and 0 neighbors w/ 5

0 - no neighbors w/ 4

2 - no neighbors w/ 4

0, 2 - each have 1 neighbor w/ 1

Does 35 shatter 4 or 1? No

3 - 1 neighbor w/ 4, 0 neighbors w/ 1

5 - 1 neighbor w/ 4, 0 neighbors w/ 1

Does 4 shatter 1? NO (not possible)

4 - 1 neighbor w/ 1

∴ No more children nodes possible

c.) [0, 1, 2, 3, 4, 5]

6. See Word document