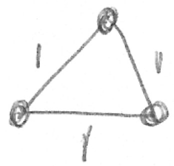
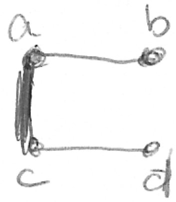
I. Examples and Counterexamples (5 points each). Do not give proofs, but clearly write a short answer or indicate your proposed example or counterexample.

1. Give an example of a weighted simple graph that has at least three distinct minimum spanning trees.

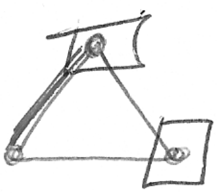


2. Give an example of a simple graph G together with a *maximal* matching $M \subseteq E(G)$ in that graph such that M is not a *maximum* matching.



$M = \{ac\}$

3. Give an example of a simple graph, a maximum matching $M \subseteq E(G)$, and a minimum vertex cover $Q \subseteq V(G)$ such that $|M| \neq |Q|$.



$|M| = 1$

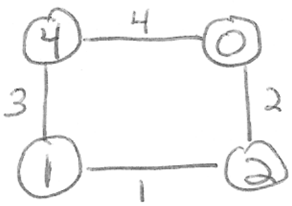
$|Q| = 2$

4. Give an example of a tree T for which the size $\alpha'(T)$ of a maximum matching is smaller than the size $\beta'(T)$ of a minimum edge cover.

$|M| = \alpha'(T) = 1$

$|L| = \beta'(T) = 2$

5. Find a graceful labeling of C_4.

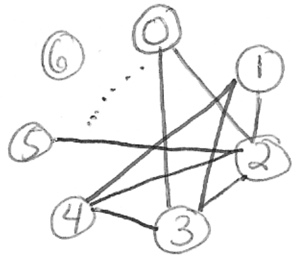


6. Label the vertices of K_7 $\{0, 1, 2, 3, 4, 5, 6\}$. Find a decomposition of K_7 into copies of the path P_4 with 4 vertices and 3 edges. Label each copy of P_4 in K_7 by writing down the list of 4 vertices of the path in adjacency order $abdc$, where $a, b, c, d \in \{0, 1, 2, 3, 4, 5, 6\}$, and $a \leftrightarrow b \leftrightarrow c \leftrightarrow d$.

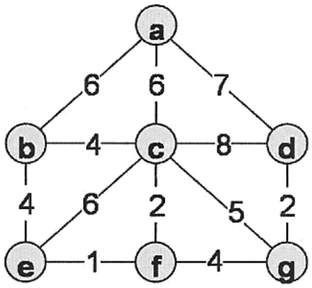
Graceful labeling of P_4:

7 translates mod 7:

3021
4132
5243
6354
0465
1506
2610

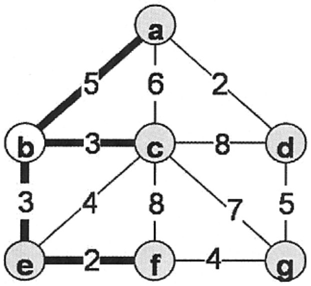


7. Give one possible order in which the vertices of this weighted graph are explored by Primm's algorithm initialized at vertex d.



d, g, f, e, c, b, a

8. Dijkstra's Algorithm is in the middle of processing the graph here. It was initialized at vertex b. In order, what are the next two edges to be added to the shortest-path tree?

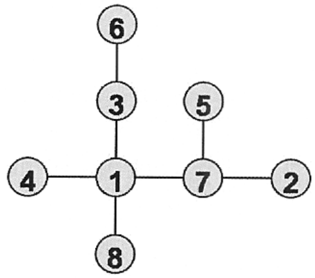


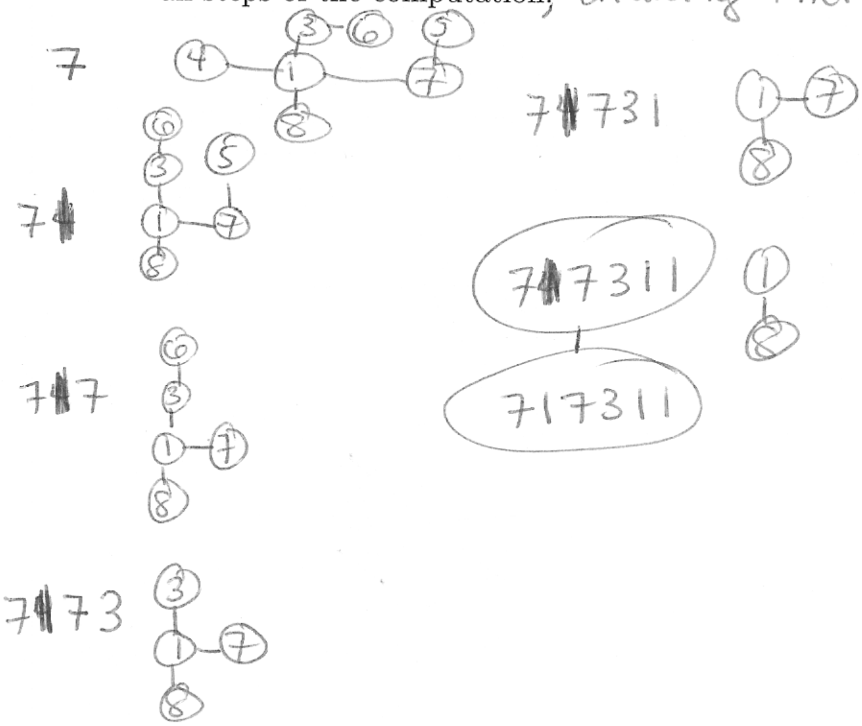
ad, fg

II. Constructions and Algorithms (15 points each). Do not write proofs, but do give clear, concise answers, showing the steps of any process or algorithm used.

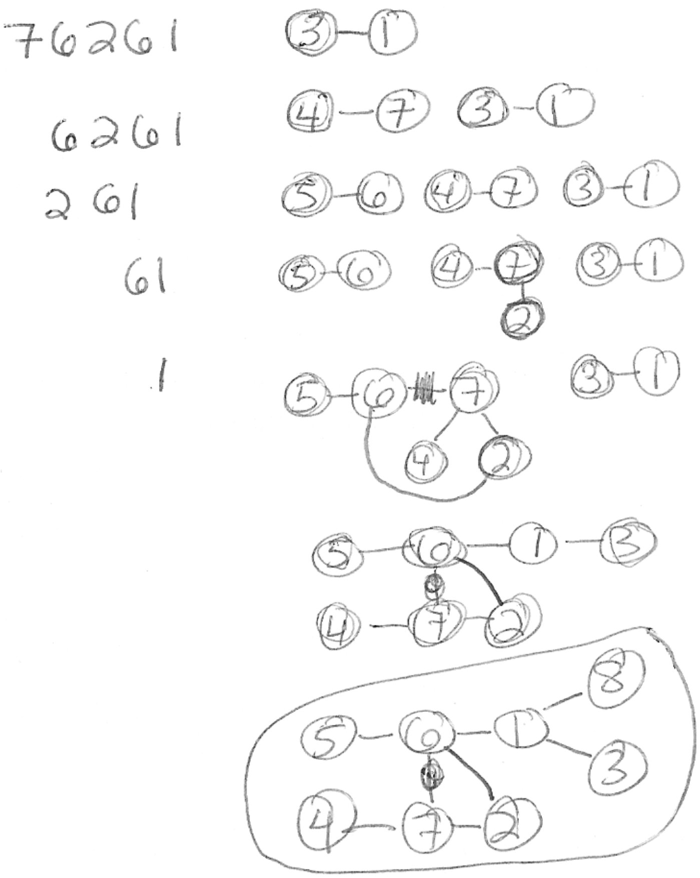
9. Prüfer Codes.

(a) Compute the Prüfer code corresponding to the tree depicted here to the right. Show all steps of the computation, drawing intermediate structures.





(b) Compute the tree with vertex set $[8] = \{1, \ldots, 8\}$ that has Prüfer code 176261. Show all steps of the reconstruction, drawing intermediate structures.



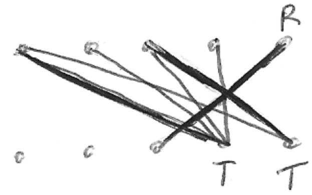
10. Use the Hungarian Algorithm to find both a maximum weight matching and a minimum cost cover in the weighted bigraph represented by the matrix below. For full credit, you must show all steps of the procedure and write down and label all intermediate structures.

$$\begin{array}{c|c}
u\backslash v & \begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \end{array} \\
\begin{array}{c} 8 \\ 7 \\ 9 \\ 6 \\ 7 \end{array} & \begin{bmatrix} 7 & 2 & 3 & 8 & 8 \\ 6 & 1 & 3 & 7 & 1 \\ 5 & 2 & 6 & 9 & 9 \\ 2 & 5 & 3 & 6 & 2 \\ 5 & 1 & 7 & 4 & 3 \end{bmatrix}
\end{array}$$

excess

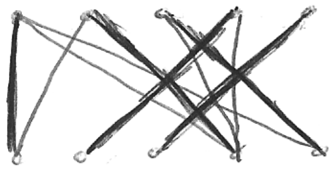
	0	0	0	0	0	
8	1	6	5	0	0	
7	1	6	4	0	6	
9	4	7	3	0	0	
6	4	1	3	0	4	
7	2	6	0	3	4	R
				T	T	

$\varepsilon = 1$



excess

u\v	0	0	0	1	1
7	$\boxed{0}$	5	4	0	0
6	0	5	3	$\boxed{0}$	6
8	3	6	2	0	$\boxed{0}$
5	3	$\boxed{0}$	2	0	4
7	2	6	$\boxed{0}$	4	5



$(w(M) = c(u,v) = 35)$

III. Proofs (10 points each). Partial credit for setting up a good proof structure without completing a proof.

11. Let G be a simple graph with *distinct* nonnegative edge weights. Prove that G has exactly one minimum spanning tree using a proof by contradiction with the following outline: if G has two distinct minimum spanning trees, then the edge weights are not all distinct. (Fill in the details of the proof.)

Proof (contradiction) Let T, T' be distinct MST's of G.

Let $e \in E(T) \setminus E(T')$.

$T'+e$ forms exactly one cycle in T'.

There are exactly two components of $T-e$.

The cycle C in $T'+e$ has vertices from both of those components, and goes from one component to the other on e, and so must go back to the original component on some edge $f \in C-e$.

$T-e+f$ is a tree, since f connects the two components of $T-e$.

$T'+e-f$ is a tree, since f is not a cut-edge in $T'+e$.

$w(e) \neq w(f)$ is impossible, since then one of the new trees would have lower weight. Therefore $w(e)=w(f)$. $\times\!\!\times$

12. Suppose a tree T has n vertices, and the maximum size of an independent set in T is $\alpha(T)$. Express $\beta(T)$ and $\alpha'(T)$ in terms of n and $\alpha(T)$, and justify your answer. (Hint: make up some example trees if you don't know where to start.)

$S \subseteq V(T)$ is an independent set in T

iff $\bar{S} \subseteq V(T)$ is a vertex cover.

Therefore ~~max~~ $\alpha(T) = \max\{|S| : S \text{ indep set in } T\}$

$= \max\{n - |\bar{S}| : \bar{S} \text{ vertex cover in } T\}$

$= n - \min\{|\bar{S}| : \bar{S} \text{ vertex cover in } T\}$

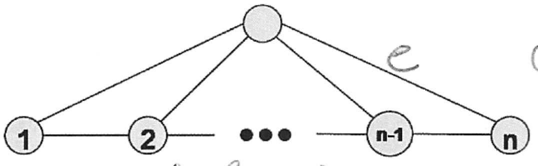
$= n - \beta(T)$.

T bipartite $\Rightarrow \alpha'(T) = \beta(T)$.

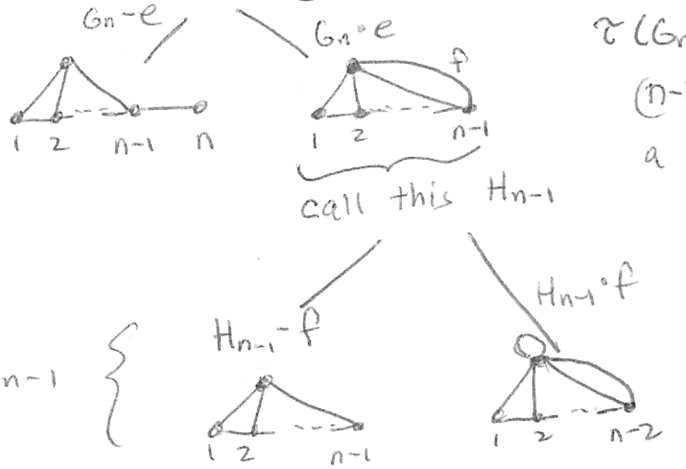
Therefore $\beta(T) = n - \alpha(T)$

$\alpha'(T) = \beta(T) = n - \alpha(T)$.

13. Let G_n be the graph shown. Find a recurrence relation for the number of spanning trees $\tau(G_n)$, and give enough initial conditions so that all values can be found. Carefully format your work so that every step is both justified and readable.



call this graph G_n.



$\tau(G_n - e) = \tau(G_{n-1})$ since $(n-1)n$ must be an edge in a spanning tree of $G_n - e$.

$H_{n-1} - f = G_{n-1}$

$\tau(H_{n-1} \circ f) = \tau(H_{n-2})$ since loops may not be in spanning trees.

$$\tau(G_n) = \tau(G_{n-1}) + \tau(H_{n-1}) \qquad (*)$$
$$= \tau(G_{n-1}) + \tau(G_{n-1}) + \tau(H_{n-2}) \qquad (**)$$

now use $(*)$ with n replaced by $n-1$:

$$\tau(G_{n-1}) = \tau(G_{n-2}) + \tau(H_{n-2})$$

solve for $\tau(H_{n-2})$, plug into $(**)$:

$$\tau(G_n) = 3\tau(G_{n-1}) - \tau(G_{n-2}).$$

Initial conditions: $\tau(G_1) = \tau(|) = 1$
$\tau(G_2) = \tau(\triangle) = 3.$