Exam 3 – Math 215

Problem 1 (35 points; 5 points each)**.** Decide if each of the following are true or false. You do not need to justify your choice here.

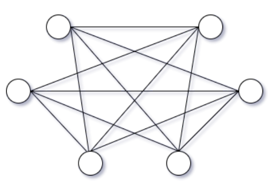
(a) FALSE Any non-planar graph contains an isomorphic copy of K_5 or $K_{3,3}$.

This is false. It is true that a graph is non-planar if it contains a subgraph homeomorphic to K_5 or $K_{3,3}$.

(b) TRUE Given a simple connected planar graph G. Without drawing G, you know how many faces it has.

Yes, we know $f = 2 + e - v$

(c) TRUE There is a way to draw the following without lifting your pencil.



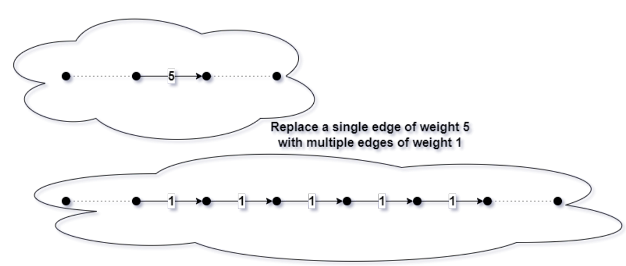
$K_{2,2,2}$

Yes $\deg(v)$ is even for all nodes v.

(d) TRUE $K_{2,2,2}$ has a Hamiltonian cycle.

(e) FALSE The diameter of $K_{2,2,2}$ is 3, that is $\text{diam}(K_{2,2,2}) = 3$ where $\text{diam}(G) = \{d(u,v) \mid u, v \text{ nodes of } K_{2,2,2}\} = 3$ and $d(u,v) =$ distance from u to v.

(f) TRUE Breadth First Search can be used to find shortest paths in connected weighted simple graphs where the weights are positive integers by replacing an edge e with weight n by a sequence of edges $e_1, e_2, \ldots, e_n$ of weight 1.



(g) TRUE Every Boolean function $F : B^n \to B$ is given by a Boolean expression in variables x_1, x_2, ..., x_n using only the operations $+$ and $^-$.

Linear combinations of solutions are solutions.

Problem 2 (Multiple Choice; 20 points; 4 points each)**.** You may select any number of choices, 0 – 4. You get one point per each correct item, meaning that if the item should be selected and you select it, then you earn a point, and if the item should not be selected and you don't select it, then you earn a point.

(a) Which algorithm could you use to find the distance between two nodes u and v of an undirected graph?

$\otimes$ Breath First Search.
◯ Depth First Search.
◯ Prim's Algorithm.
$\otimes$ Dijkstra's Shortest Path Algorithm.

(b) What algorithm could be used to find all components of a graph?

$\otimes$ Breadth First Search.
$\otimes$ Prim's Algorithm.
$\otimes$ Depth First Search..
$\otimes$ Kruskal's Algorithm.

(c) Every tree $T = (V, E)$ satisfies.

$\otimes$ T is planar.
$\otimes$ $m = n - 1$ where $m = |E|$ and $n = |V|$.
$\otimes$ T has at least two leaves.
$\otimes$ T is bipartite.

(d) In a directed graph let $d(u, v)$, the distance from u to v, be the length of the shortest path from u to v. By definition $d(u, u) = 0$ for all u and $d(u, v) = \infty$ when there is no path from u to v. Which of the following are true?

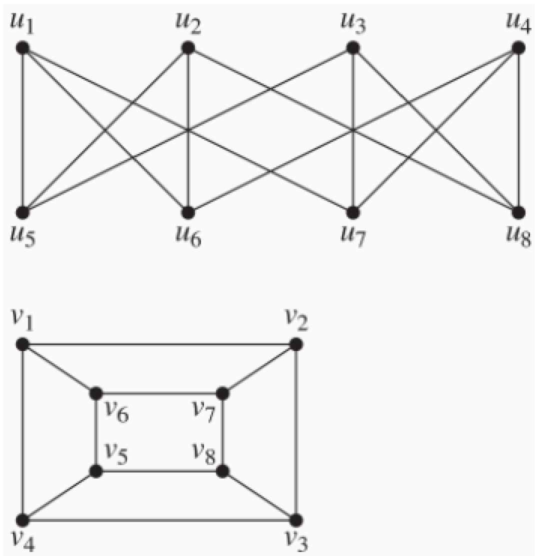
◯ $d(u, v) = d(v, u)$ (symmetry)
$\otimes$ $d(u, v) + d(v, w) \geq d(u, w)$. (triangle inequality)
◯ If $d(u, v) < \infty$, then $d(v, u) < \infty$.
$\otimes$ $d(u, v)$ can be found using BFS.

(e) Recall that the dual of a Boolean function $F : B^n \to B$ is the function $F^d : B^n \to B^n$ defined by $F^d(x_1, \ldots, x_n) = \overline{F(\bar{x}_1, \ldots, \bar{x}_n)}$. A function is self-dual iff $F = F^d$. Which of the following are self-dual?

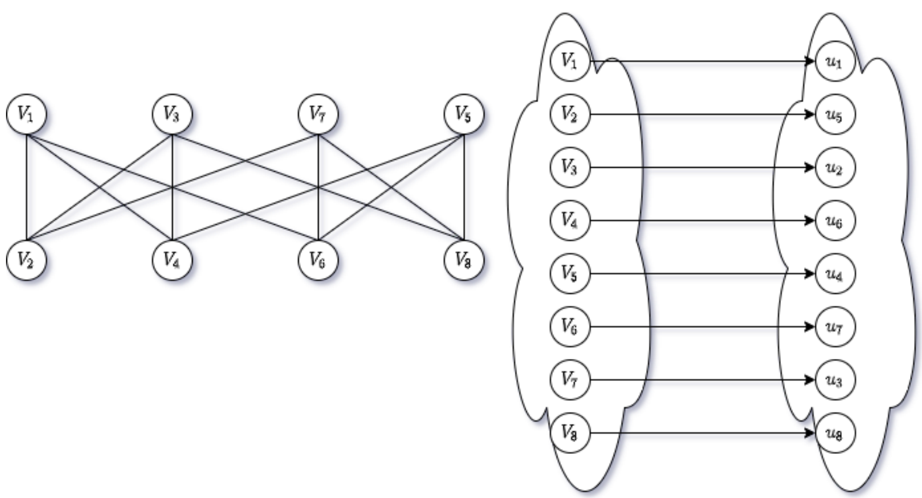
$\otimes$ $F(x, y) = x$
◯ $F(x, y) = xy + \bar{x}\bar{y}$
◯ $F(x, y) = x + y$
$\otimes$ $F(x, y) = xy + \bar{x}y$.

Problem 3 (Computation; 75 points; 15 points each). Choose **five** of the six problems, I will grade the first five that you attempt, so make it clear which you want graded.

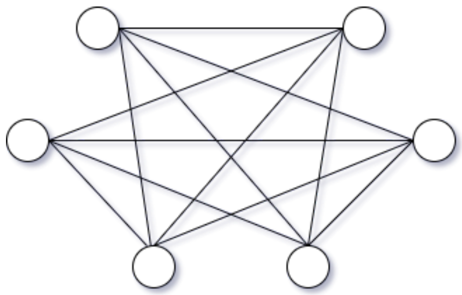
(a) Either show that the following are isomorphic graphs by providing a bijection between the vertices that preserves adjacency, or else argue that these are not isomorphic by giving some property that one graph has, but the other does not.



Here is the isomorphism:



(b) Determine if $K_{2,2,2}$ is planar or not and prove your claim.

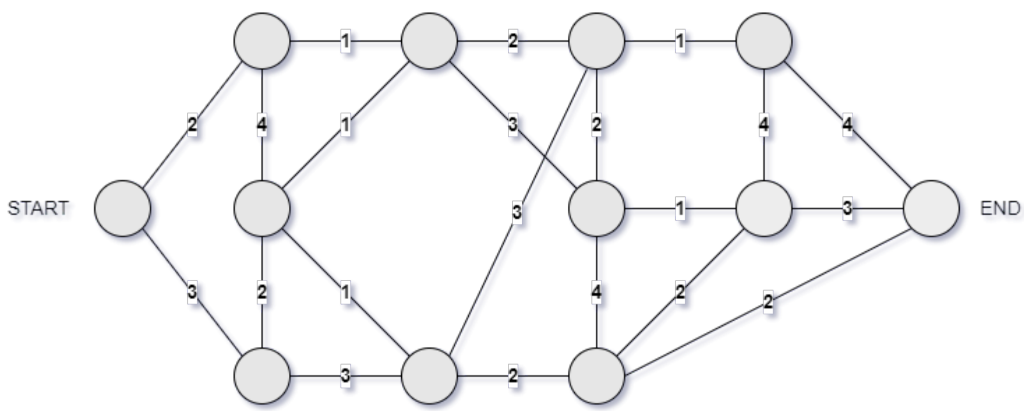


$K_{2,2,2}$

There are two ways to proceed. First, just find a planar representation (See here.)

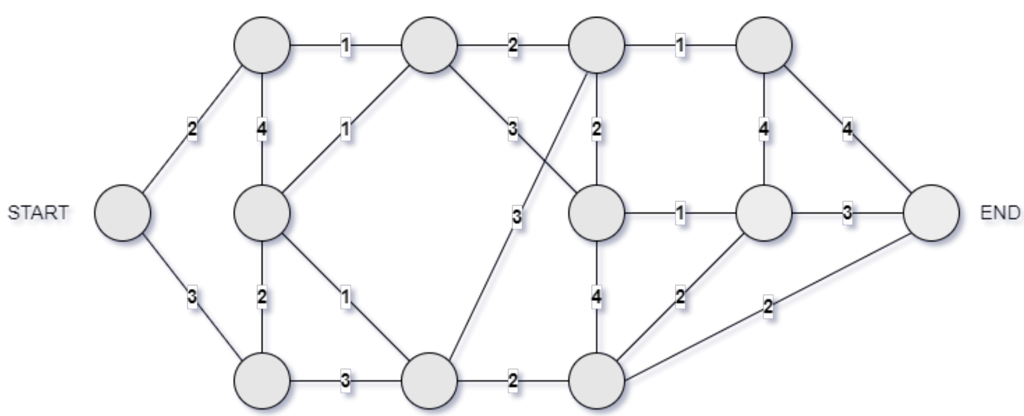
Or you can use Kuratowski's Theorem. At the moment, I don't see a very easy way to pull this off, so if you tried this, I will have to see what you did.

(c) Use Dijkstra's algorithm to find the shortest path From the START node to the END node. Make sure that I can see how you are using Dijkstra, what the weight of the shortest path is as well as what the path itself is.



See here.

(d) Use either Prims's or Kruskal's algorithm to find a minimal spanning tree. Make sure that I can tell what algorithm you have chosen and can see how you are using the algorithm. Make clear what the weight of the minimal spanning tree is, as well as what the tree itself is.



For Prim's see here. For Kruskal's see here.

(e) Build the binary search tree for the words in the sentence "The quick brown fox jumps over the lazy hound dog." Make sure to show how the tree was constructed.

See here.

(f) Sum-of-products expression E in x, y, z, w so that

$$F_E(x, y, z, w) = 1 \text{ iff } (z = w \text{ and } x \neq y) \text{ or } (z \neq w \text{ and } x = y)$$

x	y	z	w	F(x,y,z,w)
1	0	0	0	1
0	1	0	0	1
1	0	1	1	1
0	1	1	1	1
0	0	1	0	1
0	0	0	1	1
1	1	1	0	1
1	1	0	1	1

$$E = x\bar{y}\bar{z}\bar{w} + \bar{x}y\bar{z}\bar{w} + x\bar{y}zw + \bar{x}yzw + \bar{x}\bar{y}z\bar{w} + \bar{x}\bar{y}\bar{z}w + xyz\bar{w} + xy\bar{z}w$$

Problem 4 (Theory/Proofs; 40 points; 20 points each)**.** Select **two** of the following four to complete. As above, you must make clear which three you choose.

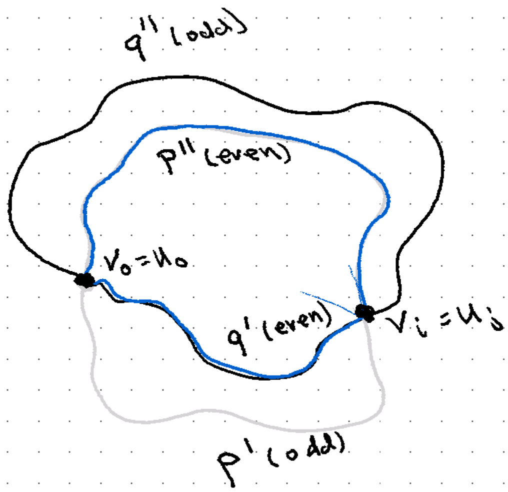
(a) Show that if G is a connected simple weighted graph and T is a minimal spanning tree, then for any edge of G that is not in T, there is an edge e' in T so that $w(e') \leq w(e)$.

Since e is not in T and T is spanning, $T + e$ has a cycle. If $w(e) < w(e')$ for any e' in the cycle other than e, then $S = T - e' + e$ is a spanning tree and $w(S) < w(T)$. Thus $w(e') \leq w(e)$ for all e' in the cycle. Since there is some e' in the cycle, we have the desired e'.

(b) Call a simple connected graph a **cactus** iff no simple cycle in the graph shares an edge with another simple cycle. Show that a graph all of whose simple cycles are of odd length is a cactus. (It is easy to see that there are cacti with even length simple cycles, so this is not a characterization of cacti.)

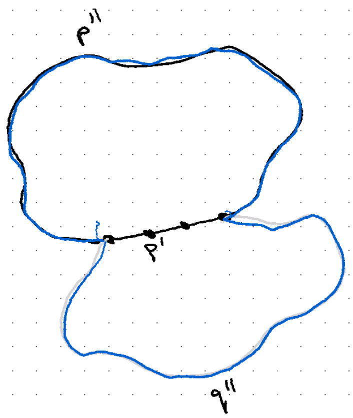
Assume all simple cycles have odd length. Towards a contradiction suppose that G is not a cactus. This means that there are two simple cycles p and q sharing an edge. In particular p and q share two nodes. There are two cases.

Case 1: Let $p = u_0 u_1 \cdots u_k$ and $q = v_0 v_1 \cdots v_l$ be two simple paths. Suppose p and q share two nodes not in a single edge. We may assume WLOG that $u_0 = v_0$ is one of those two nodes. Let i be least so that v_i is in p, then let $u_j = v_i$. Let $q' = v_0 v_1 \cdots v_i$ and $p' = u_0 u_1 \cdots u_j$, let $q'' = v_i v_{i+1} \cdots v_l$ and $p'' = u_j u_{j+1} \cdots u_k$ so $p = p' + p''$ and $q = q' + q''$ where summation of paths is defined in the obvious way. For a path r let r^{-1} be r in reverse, for example, $p'^{-1} = u_j u_{j-1} \cdots u_0$. Let $|p|$ be the length of the path. So what we have is $|p| = |p'| + |p''|$ is odd and $|q| = |q'| + q''|$ is odd. But also $q' + p'^{-1}$ is simple so $|q'| + |p'|$ is odd. Assume WLOG that $|q'|$ is even and $|p'|$ is odd, so $|q''|$ is odd and $|p''|$ is even.



The key is that $q' + p''$ is also simple else there is some node in q' that is in p'', this contradicts the definition of v_i. But $q' + p''$ has even length. A contradiction!

Case 2: The other option is that q and q share a subpath that is a sequence of edges so again WLOG we may assume $p = p' + p''$ and $q = p' + q''$, where p' is the subpath of common edges. But then $q'' + p''$ is simple and as above must be even. Again a contradiction.



(c) Below is a 10x10 pixel drawing, one byte (0-255) is used per pixel, so the "size" of the coding is 100x8 = 800 bits. Use Huffman coding to code the values (indicate what you are doing). What is the final size of the encoding in bits?

200	200	200	200	200	200	200
200	125	125	200	125	125	200
220	0	255	80	0	255	220
220	0	0	80	0	0	220
0	190	190	190	190	190	0
220	0	80	80	80	0	220
220	220	0	0	0	220	220

Value	Frequency	Relative Frequency
255	2	0.04
125	4	0.08
80	5	0.10
190	5	0.10
220	10	0.20
200	10	0.20
0	13	0.27

Here is a solution

200	200	200	200	200	200	200
200	125	125	200	125	125	200
220	0	255	80	0	255	220
220	0	0	80	0	0	220
0	190	190	190	190	190	0
220	0	80	80	80	0	220
220	220	0	0	0	220	220

Value	Frequency	Relative Frequency	Bits	
255	2	0.04	4	8
125	4	0.08	4	16
80	5	0.10	4	20
190	5	0.10	4	20
220	10	0.20	2	20
200	10	0.20	2	20
0	13	0.27	2	26

130
Total Bits

See here for the coding.