

Problem 1: (5.3.3)

- (a) Let
- $T = (V, E)$
- be a tree.

Let's induct on the number of vertices, and prove that each tree (n vertices) has an ordering of vertices such that v_n is adjacent to v_1 and every edge v_i, v_{i+1} is of maximum distance 3 away from each other.

Base case: Assume that T has three vertices or fewer. Let's take $|V| = 3$ for example. Then the tree would be isomorphic to something like $V = \{v_1, v_2, v_3\}, E = \{\{v_1, v_2\}, \{v_2, v_3\}\}$. Take v_1 and v_2 . Then we can find the path v_1, v_3, v_2 creates an ordering on the tree that satisfies our conditions.

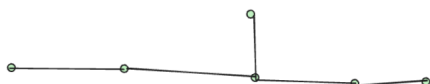
Induction Hypothesis: Assume we can find such ordering on trees of less than n vertices. Let T be a graph with $n \geq 3$ vertices.

Take two adjacent vertices, v, w . Since T is a graph with more 3 vertices, we can either find a vertex connected to either v or w . Let's say its connected to v , and call that vertex v' . If we split T by the edge $\{v, w\}$, then we get two trees, one that contains v and the other that contains w . Using the induction hypothesis, there exists a Hamiltonian Path from v to v' in the tree containing v .

Assume that w isn't a leaf. Then there exists a w' that it is connected to that isn't v . Then, in the tree containing w , there exists a hamiltonian path starting at w and ending at w' . Notice that w' and v' are a distance of 3 apart. So, there exists an edge between them in $T^{(3)}$. This results in a hamiltonian that starts at w and ends at v . Hence, we can find such hamiltonian path (and thus cycle).

If w is a leaf, then we can connect w to v' and everything works still

- (b) Since G is connected, we can find a spanning tree T of G . Applying part a, we can find a Hamiltonian Cycle on $T^{(3)}$ which is a subgraph of $G^{(3)}$.



(c)

Problem 2: (5.4.6) Let w and w' be two weight functions on $G = (V, E)$. Let's say the MST T was constructed using Kruskal's algorithm (Picking the lowest edge every time that doesn't create a cycle).

Since the if $w(v) < w(v')$ iff $w'(v) < w'(v')$, this implies the ordering of the vertices from lowest weight to greatest weight remains the same. So, we would select the same edges at each iteration of the algorithm. Hence, T would be the MST relative to w and w' .

Problem 3: (5.4.8)

- (a) Let $G = (V, E)$ and let T be the minimum spanning tree on G .

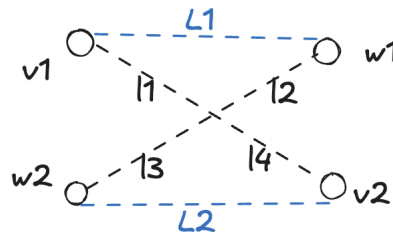
Assume for contradiction that some vertex x has degree 7 (same can be proven for greater than 7).

Let the vertices its connected to be v_1, v_2, \dots, v_7 . Then, $\text{dist}(x, v_i) \leq \text{dist}(v_i, v_j)$ for all $i \neq j$.

However, this leads to a contradiction. Notice, that the angle between some two vertices connected to x must be at least $\frac{360^\circ}{7}$. Let these two vertices be v_i and v_j . Using some basic trigonometry, if we form a triangle between x, v_i, v_j , we know the longest side in the triangle must be opposite of the greatest angle. So by assumption, the angle $\angle v_i x v_j$ is the largest angle. This means the sum of angles is at most $\frac{360^\circ}{7} \cdot 3$. However this doesn't add up to 180° , leading to a contradiction since the angles of a triangle must be 180. This means there cannot exist a vertex of degree 7 (or more).

- (b) Let's prove that every step of creating a minimum spanning tree, that we will not cross over an existing edge.

In other words let's show that $v_1 v_2, w_1 w_2$ will never occur.



By the triangle inequality, it is evident that $L_1 + L_2 \leq l_1 + l_2 + l_3 + l_4$. So the only way that picking $l_1 + l_4, l_2 + l_3$ might occur is if there is a cycle that prevents the the greedy algorithm from picking L_1, L_2 .

Let's assume that v_1 and w_2 are connected, and that the algorithm already chose the edge v_1, v_2 .

Let's assume that $L_1 \leq l_2 + l_3$ (and obviously $L_1 \leq l_1 + l_4$). Then we choose L_1 and there will be no intersections. Otherwise, $L_1 > l_2 + l_3$. At the step we choose v_1, v_2 we know that $l_1 + l_4 < L_2$. So, $l_1 + l_2 + l_3 + l_4 < L_2 + L_1$. This must be a contradiction. Hence, this cannot occur and there will crossings in the found MST.

Problem 4: (5.4.11) Let's show that the greedy edge covering is at most $\frac{3}{2}$ the size of the smallest edge covering. Let $G = (V, E)$ be a connected graph where $|V| = n$.

Let E_M be the edge set of the minimum edge covering. Let E_G be the edge set of the edge covering of the greedy algorithm.

Let E'_G be the edge set after first step of the algorithm (after picking a matching of G). let E'_M be an edge covering constructed by removing edges E_M until we get a matching of G (that is maximal?).

Note that we know that $|E_M| \leq |E_G|$.

Claim 1: We claim that $|E'_M| \geq |E'_G|$. Intuitively, this must be true because we want to add less edges to E_M to get the inequality $|E_M| \leq |E_G|$.

Claim 2: $|E'_G| \geq \frac{1}{2}|E'_M|$. Take an edge in E'_G . Either it is in E'_M or at least one of its vertices are covered by E'_M (otherwise we could add the edge to E'_M).

So we get the inequality:

$$|E'_M| \geq |E'_G| \geq \frac{1}{2}|E'_M|$$

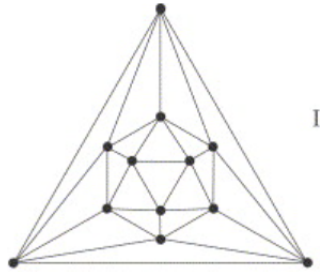
Using this we can show that $\frac{|E_G|}{|E_M|} \leq \frac{3}{2}$.

Note that $n - |E'_G| = |E_G|$ because we are adding $n - 2|E'_G|$ edges to the remaining $n - 2|E'_G|$ vertices. Also note that $|E'_M| \leq \frac{n}{2}$ because there are only n vertices in G . Similarly, $n - |E'_M| = |E_M|$. We know that $n - |E'_G| \leq n - \frac{1}{2}|E'_M|$ from the claim above.

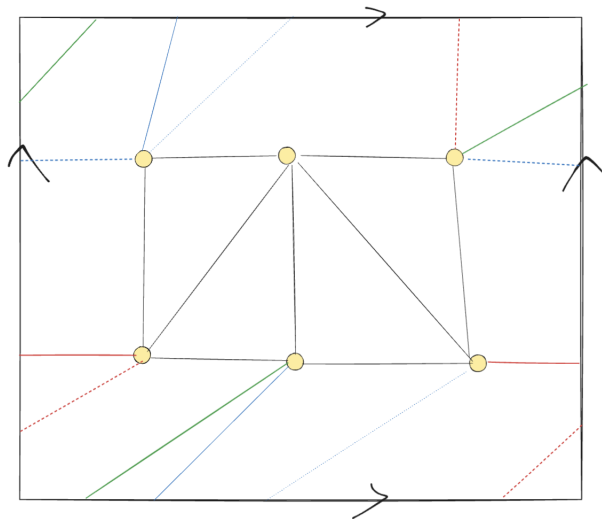
Hence,

$$\begin{aligned} \frac{|E_G|}{|E_M|} &= \frac{n - |E'_G|}{n - |E'_M|} \leq \frac{n - \frac{|E'_M|}{2}}{n - |E'_M|} \\ 2|E'_M| &\leq n \\ \Rightarrow 2n - |E'_M| &\leq 3n - 3|E'_M| \\ \Rightarrow \frac{n - \frac{|E'_M|}{2}}{n - |E'_M|} &\leq \frac{3}{2} \\ \Rightarrow \frac{|E_G|}{|E_M|} &\leq \frac{3}{2} \end{aligned}$$

Problem 5: (6.1a) This shape



Problem 6: (6.2b)



Problem 7: (6.3.1) We know that in a maximally planar graph, an edge bounds exactly two faces. Moreover, since each face cannot be bounded by a triangle, the cycle the bounds the face must be at least 4 edges.

So we get the inequality,

$$2|E| = \sum_F \text{edges per face} \geq 4|F|$$

This means $|E| \geq 2|F|$. By Euler's formula, we know that $2 = |V| - |E| + |F|$. So,

$$\begin{aligned} 2 &= |V| - |E| + |F| \leq |V| - |E| + \frac{|E|}{2} \\ \Rightarrow 2 + \frac{|E|}{2} &\leq |V| \\ \Rightarrow |E| &\leq 2|V| - 4 \end{aligned}$$

Problem 8: (6.3.8a, c)

- (a) We know that a vertex can have at max degree 3. So, at the start, we have $3n$ possible places to connect the two vertices. At each step, we "remove" two possible connections by connecting two vertices, before adding one back. So, each step yields a net -1 places to draw a connection. Thus, it appears that we can have $3n$ turns before we run out of possible places to draw the arc. However, we can't perform the last step, because we need to have two available vertices to begin with before adding one back. Hence, we can only take $3n - 1$ turns.
- (b) Notice that the game ends when there is at most one arrow pointing into a face. Also, note that there are $4n$ possible places to draw an arc, and each step doesn't decrease the potential places to draw an arc, since we are creating two more nodes/arrows.

So, we can perform a maximum $n - 1$ turns before changing the number of faces present in the graph (connecting all n nodes). After this, each step creates exactly one additional face. Since we start out with 1 face, then it will take at max $4n - 1$ more turns to create $4n$ faces, and at this point the game must end because there are only $4n$ arrows.

Adding these two values results in a maximum of $5n - 2$ turns.