Alec Xiang

Math 244, Midterm #2

Problem 1.

First create an empty set V_T of the vertices in our spanning tree T. At each step, find the vertex v in G of minimum weight and add it to V_T. For each of v's adjacent vertices in G, add those adjacent vertices to V_T and the edges between those adjacent vertices and v to T, if they have not already been added.

We will never have a v that is greater weight than all adjacent vertices and is not already in T. If we have a v that is less weight than all adjacent vertices, but all adjacent vertices are already in T, add v to V_T by connecting it with the one smallest adjacent vertex (so that we will not have a cycle ever as we will never join the "key" vertex to two sides of a "partial" cycle). Eventually, all vertices will be added to V_T s.t. $\sum_{v \in V} deg_T(v)\, w(v)$ is minimized.

Verifying this greedy algorithm, we can view $\sum_{v \in V} deg_T(v) w(v)$ in terms of edges. That is, each edge in T has two vertices as endpoints, and this is a sum over the weights of those endpoints. Our algorithm ensures that each vertex is included among this set of edges in a minimal way, because a certain vertex in an edge was either the minimum at that step, or added to V_T with its minimum neighboring vertex.

Problem 2.

a) $ex(n, K_{1,3}) = n$ for $n \geq 3$

Edge cases are when n=1 and n=2, when we have 0 and 1 edges respectively.

Otherwise, we can configure the points in a cycle of length n, making for n edges. We can't have any more edges, because if we draw another edge between any two vertices, we will have a $K_{1,3}$. The other way we can get to n edges is creating many smaller cycles or "rings", as long as each vertex has degree 2.

b) If we can't have a $2K_2$, that means we can't have a pair of disjoint edges. So, every pair of edges must share a vertex. If we have n vertices and the shared vertex is v, the degree of v must be at most $n-1$, meaning $ex(n, 2K_2) = n-1$.

We can show that we can't add any more edges to this graph, which has v in the middle and $n-1$ edges attached to v and another vertex. If we add an edge without adding any more vertices, it must be an edge between two vertices that are both not v. This new edge will then form a $2K_2$ with an edge between v and an edge that is not one of those two edges.

Our formula holds true for n=1 and n=2, but a special case we have to consider is when $n=3$ and our extremal graph is a triangle. In this case, when we initially build a graph with v which has 2 attached vertices, we can in fact connect those two other vertices to each other without forming a $2K_2$ because any two edges in this triangle must share a point. Thus, for the special case n=3, $ex(n, 2K_2) = n$.

To summarize, in general the graph has one central vertex with all edges shooting out from it. And if n=3, we just have a triangle.

Problem 3.

a) $cr(K_{2,2,2}) = 0$

Let the 3 parts of the tripartite graph be A, B, and C. We can draw a cycle of length 4 as a square, with the points at diagonals to each other in A and B.

We can draw C in our graph as follows. Both the points in C have to be connected to all the points of the square without being connected to each other to satisfy the tripartite property. We can do this by putting one of C's points inside the square, and one outside the square. We can then draw edges between C's points and the square without any intersections or an edge between C's points.

b) $cr(K_6) = 3$

Let us consider K_5, which we know must have at least one edge crossing. We can put edges in the interior and exterior of 5 in 3 unique ways, e.g. 0, 1, or 2 edges on the outside (for 3, 4, and 5 we can simply swap what we mean by interior and exterior). For 0 edges on the outside, we have 5 interior crossings. For 1 edge on the outside, we have 3 interior crossings. For 2 edges on the outside, we have 1 interior crossing, so we will proceed with this case.

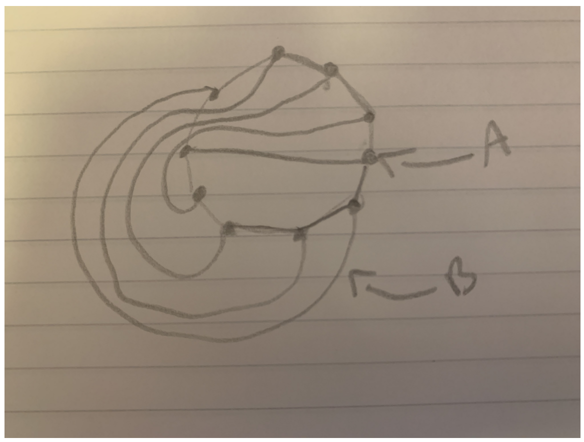
If we put the 6th point on the outside and connect it to all 5 points, it will cross the two exterior edges for a total of 3 crossings → $cr(K_6) = 3$

c) $cr(G_{2n}) = n-2$

We can start by drawing one edge A across the cycle's interior and one edge B offset by one from the first edge around the cycle's exterior. There are now n-2 points between A and B for which we have to draw edges a distance n apart.

All the n-2 points we have to draw edges must pass through at least 1 already placed edge (because they are "locked" inside a closed polygon on all sides). Additionally, A and B are offset by 1 to make the below drawing possible.

We can have each of these edges go between the offset of A and B, adding one to the crossing count for a total of n-2. The picture below illustrates what this would look like for n=5, where $cr(G_{10})=5-2=3$:



Problem 4.

We will prove this by contradiction. Let us assume $\chi(G) \geq 65$. That means there must be at least $\binom{\chi(G)}{2}$ edges for a coloring of $\chi(G)$, because every 2 different colors must have an edge between them. This is true because otherwise we could reduce the number of colors we use, for example, by changing all blue to red if blue is not connected to the red anywhere.

In other words, if $\chi(G) \geq 65$, then $|E| \geq \binom{65}{2} = 2080$. However, the problem tells us $|E| = 2020$, so by this contradiction we must have $\chi(G) \leq 64$.

Note: We can check this by seeing that $\binom{64}{2} = 2016$, so we can't reapply this proof to reduce the color bound further than $\chi(G) \leq 64$.

QED

Problem 5.

We'll prove this via double counting, i.e. counting the number of cherries, *T,* in *G*, where we define a cherry as having 2 children a_1 and a_2 in A and a parent in B. I'll term cycles with the product of differences $\geq$N as "forbidden cycles".

The first way we'll count cherries is considering all the possible distances between the cherry's children. There are N-1 distances (the smallest distance is 1, the largest is N-1), where we'll call each distance *d*. For cherries of each distance, we can only have N/*d* consecutive parents in B, so that we don't have 2 cherries that form a forbidden cycle. For instance, if we have parents with the same *d* farther than N/*d* from each other in B, then our product will $\geq$N forming a forbidden cycle. This gives us $T \leq \sum_{d=1}^{N-1} (N-d)\ [N/d]$ (where I'm using brackets as the floor function). We have a $\leq$ here because this first way of counting yields the maximal number of cherries (which are not all necessarily in *G*) .

Simplifying using a constant "β", we get $T \leq \sum_{d=1}^{N-1} \beta \frac{N^2}{d}$. We can further simplify this using that the sum of a harmonic series is logarithmic (and applying change of base to get a base of 2), yielding $T \leq \beta N^2 log_2 d$.

* * *

The second way we'll count cherries is by considering all possible cherries for each vertex in B. This is simply $T = \sum_{i=1}^{N} \binom{deg_b i}{2}$, as we are picking 2 children from all the edges attached to the vertex in B. Using that $\binom{n}{2} = n(n-1)/2$ and a constant "α", we simplify this to $T = \sum_{i=1}^{N} \alpha deg_b i$. Using that the sum of degrees is twice the number of edges and applying Cauchy-Schwarz, we get $T \geq \alpha \frac{(2|E|)^2}{N}$.

* * *

Combining our two ways of counting, we get $\alpha \frac{(2|E|)^2}{N} \leq \beta N^2 log_2 d$. Now we will use a new constant "c". Simplifying into this constant, we get $(|E|)^2 \leq cN^3 log_2 d$. Taking the square root of both sides, we have the inequality we are trying to prove: $|E| \leq cN^{3/2} (log_2 d)^{1/2}$

QED