17.1 Extremal Coloring

We'll now examine the structure of k-chromatic graphs, or graphs with chromatic number of k.

How small can a k-chromatic graph be? We can show that every k-chromatic graph with n vertices has at least $\binom{k}{2}$ edges.

How large can a k-chromatic (simple) graph be? Let's think in terms of multipartite graph. A **complete multipartite graph** is a generalization of a complete bipartite graph for some arbitrary number x independent sets $\{S_1, S_2, \ldots, S_x\}$, where $\forall u, v \in V(G) : (u, v) \in E(G)$ iff $u \in S_i, v \in S_j : i \neq j$. Obviously, a complete multipartite graph is k-chromatic when there are k sets. A **Turán Graph** is the complete r-partite graph with n vertices whose partite sets differ in size by at most 1, i.e., they have sizes of either $\lfloor \frac{n}{r} \rfloor$ or $\lceil \frac{n}{r} \rceil$.

Among simple r-partite graphs with n vertices, the Turán graph is the unique graph with the most edges. Further, among n vertex graphs with no $r+1$-clique, the Turán graph has the maximum number of edges. In other words, A Turán graph is the maximal n vertex graph with an r-coloring.

17.2 Color-critical Graphs

Remember that a graph G is **color-critical** when every subgraph H of G has a lesser chromatic number. This implies that the removal of any edge or vertex from G decreases the minimal number of colors required for a proper coloring, or $\chi(G - e) < \chi(G), \forall e \in E(G)$. To show a graph is color-critical, we only need to compare it with subgraphs obtained by removing a single edge.

For a k-critical graph, we can say that there exists on a proper k-coloring of G, $\forall v \in V(G)$ the color on v appears nowhere else and the other $k-1$ colors appear in $N(v)$. Additionally, $\forall e \in E(G)$, every proper $k-1$ coloring of $G-e$ gives the same color to the two endpoints of e.

If G is a graph with $\chi(G) > k$ and has partitions X, Y, where $G[X]$ and $G[Y]$ are k-colorable, then the edge cut $[X, Y]$ has at least k edges. In addition, every k-critical graph is $(k-1)$-edge-connected

17.3 Subdivions and coloring

An S-**lobe** of some graph G with vertex set $S \subseteq V(G)$ is an induced subgraph on G with vertices in S and some component of $(G - S)$. Note that for a given S there can be multiple S-lobes. We'll use this concept in a few current and later proofs.

If G is a k-critical graph, then G has no vertex cut consisting of pairwise adjacent vertices. In other words, if G has vertex cut $S = \{x, y\}$, then $x \notin N(y), y \notin N(x)$ and G has an S-lobe H such that $\chi(H + (x, y)) = k$.

Recall an edge **subdivision**, which takes edge $e = (u, v)$ and splits it into $f = (u, w), g = (w, v)$, where u, v otherwise retain their original adjacencies and w is some new vertex. Note that each of f and g can be further subdivided. We can also consider subdivisions into terms of graphs and subgraphs. An **H-subdivision** is a graph obtained from (sub)graph H by subdividing some subset of H's edges some arbitrary number of times.

We can use the above concepts to prove how every graph G with a chromatic number $\chi(G) \geq 4$ contains a K_4-subdivision. We'll get to explore similar ideas in coming lectures.

17.4 Minimum Vertex Coloring

In the context of vertex coloring, we've focused on placing bounds on the chromatic number of a graph. Next class, we'll discuss how it might be possible to determine the chromatic number $\chi(G)$ exactly. The **minimum vertex coloring** problem is the problem of coloring a graph G with $\chi(G)$ colors, or the minimum number of colors possible. This problem is NP-complete. Solving it exactly in the general case is exponential in the size of the graph, with known approaches being backtracking/dynamic programming or just brute force enumeration.

As we've noted with greedy coloring, we have no guarantee that a minimum number of colors will be output by the algorithm. As we can also observe, modifying the processing order of vertices will change the quality (in terms of number of colors) of the end result. In fact, there does exist some processing ordering of vertices such that greedy coloring's output will be optimal (homework problem). Unfortunately, determining such an ordering is NP-hard.

In general, the minimum vertex coloring problem is usually tackled with heuristics. One of the more well-known is from **Brélaz**, where we process vertices in order of which vertex currently has the most colors in its neighborhood. Ties are broken based on which vertex has the most uncolored vertices in its neighborhood. The rationale for this order is that we first color the vertices which are the most "difficult" to color.