22.1 Line Graphs

The **line graph** of G, written as $L(G)$, is the simple graph whose vertices are the edges of G. Edges $v, u \in E(G)$, represented as vertices $v, u \in V(L(G))$, have an edge between them in $L(G)$ if they share a common endpoint $w \in V(G) : \{(v, w), (u, w)\} \in E(G)$. There is a relationship between problems involving edges in G and problems involving the vertices in $L(G)$:

1. An Eulerian Circuit in G is a spanning cycle in $L(G)$.
2. A matching in G is an independent set in $L(G)$.
3. A cut edge $e = (u, v)$ in G is a cut vertex in $L(G)$ if $d(u), d(v) > 1$.
4. Edge-coloring in G is equivalent to vertex coloring in $L(G)$.

Consider some H such that $L(H) = G$, An interesting notion is that we can theoretically exploit this relationship. Note that it's possible to compute a maximum matching in polynomial time in general on H while a maximum independent set requires exponential time in general on G; however, we observe that there is effectively a 1-to-1 correspondence between these two problems. Likewise, the same might be said of finding an optimal vertex coloring in general takes exponential time while an optimal edge coloring only requires polynomial time. We going to focus on the last problem today. Edge-coloring in G and how it relates to vertex coloring in $L(G)$.

22.2 Edge-coloring

Edge coloring is the problem of assigning labels, i.e. *colors*, to all $e \in E(G)$ such that no two edges have the same color if they share an endpoint $v \in V(G)$. We use similar terminology as with vertex coloring. A coloring is **proper** if it satisfies the above criteria. We consider a k-**edge-coloring** to be a proper edge coloring of k colors. The **edge-chromatic-number** or **chromatic index**, $\chi'(G) = k$, is equal to the smallest k for which G is properly k-edge-colorable. Let's consider some bounds on $\chi'(G)$.

Since all edges incident on the largest degree vertex require separate colors, obviously $\chi'(G) \geq \Delta(G)$.

If we consider a greedy scheme to color edges and note that no edge shares endpoints with more than $2\Delta(G) - 1$ edges, we have the bound $\chi'(G) \leq 2\Delta(G) - 1$.

As a greedy edge coloring scheme on G is equivalent to a greedy vertex coloring scheme on $L(G)$, we further have the bounds $\chi'(G) = \chi(L(G)) \leq \Delta(L(G)) + 1 \leq 2\Delta(G) - 1$.

If G is bipartite, we can show that $\chi'(G) = \Delta(G)$.

For any simple graph, we can further show that $\chi'(G) \leq \Delta(G) + 1$. Or combined with our lower bound, $\chi'(G) = \Delta(G)$ or $\chi'(G) = \Delta(G) + 1$.

22.3 Forbidden Subgraphs

Consider the problems discussed above with equivalent representations between G and line graph $L(G)$. Let's say that we wish to identify a maximum independent set on a general graph. As stated above, computing a maximum independent set is of exponential complexity, while a maximum match can be done in polynomial time. So, we can potentially simplify our problem if we're able to identify some graph H such that G is the line graph of H, or $L(H) = G$. If we can do that, then we can solve a maximum match on H and easily translate the solution to G.

Obviously, such an H is not going to exist for all graphs, otherwise that would imply we can solve a NP-hard problem in polynomial time. The question then becomes, for what conditions does there exist such a H? Below, we're going to characterize some conditions we have of G such that a corresponding H is guaranteed to exists.

For a simple graph G, there is a solution to $L(H) = G$ if and only if G decomposes into complete subgraphs, with each vertex of G appearing in at most two of these complete subgraphs.

A **double triangle** is an induced subgraph of graph G that consists of two triangles sharing an edge and no edge existing between the vertices that comprise the third vertex of each triangle. A triangle T is **odd** if $\exists v \in V(G) : |N(v) \cap V(T)|$ is odd. For a simple graph G, there is a solution to $L(H) = G$ if and only if G is claw-free and no double triangle of G has two odd triangles.

The graphs below are a list of all **forbidden subgraphs**. For a simple graph G, there is a solution to $L(H) = G$ if and only if G does not contain any forbidden subgraph as an induced subgraph. Being able to identify the graph H in $L(H) = G$ can be done in linear time. However, a discussion of such an algorithm is beyond the scope of the course.