Draft of crib sheet for final

The material on these sheets will appear on the cover to the final exam for MCS-236. Although I have not deliberately included any errors, there may be some. It is your responsiblity to be aware of any errors on the sheet come exam time. It is also your responsibility to type up any other brief notes you would have me add to the sheet if you find something is missing.

Problems you have seen — homework problems, exam problems, reading prep problems and in-class problems — will appear on the exam. There will also be new problems. The exam is **closed book.**

Mathematical background The following concepts should be committed to memory:

Sets: $A \subseteq B$, empty set $\{\}$, partition, $A \cap B$, $A \cup B$, $A - B$, $\overline{A}$, disjoint sets, ordered pairs and k-tuples, $A \times B$.

Proofs: $\forall$, $\exists$, negating $\forall$ and $\exists$, $\wedge$, $\vee$, $\Leftrightarrow$, $\Rightarrow$, direct proof, proof by contradiction, proof by contrapositive.

Induction: Strong induction and top-down induction.

Functions: Bijection, one-to-one correspondence, inverse, composition.

Counting: Knowing when to multiply and when to add. Fluency with $n!$ and $\binom{n}{k}$.

Relations: What's a relation, equivalence relations.

Basic definitions

Graph: A vertex set $V(G)$, edges set $E(G)$ and a relation that associates with each edge two vertices. A **simple graph** has no loops or multiple edges.

Digraph: A vertex set $V(G)$, an edge set $E(G)$, and a function assigning each edge an ordered pair of vertices. A **simple digraph** has no multiple edges (but may have loops.)

The **complement** $\overline{G}$ of a simple graph G has vertex set $V(G)$ and uv is an edge in $\overline{G}$ if and only if it is not an edge in G. A **clique** is a set of pairwise adjacent vertices and an **independent set** is a set of pairwise nonadjacent vertices. (A single vertex is both a clique and an independent set.) A **subgraph of** G is a graph H so that $V(G) \subseteq V(H)$ and $E(G) \subseteq E(H)$. A graph G is **connected** if every pair of vertices belongs to a path in G. Two simple graphs, G and H, are **isomorphic** if there is a bijection $f : V(G) \to V(H)$ such that $uv \in E(G)$ if and only if $f(u)f(v) \in E(H)$.

P_n is a path with n vertices.
C_n is a cycle on n vertices.
K_n is the **complete graph** on n vertices.
K_{mn} is the complete bipartite graph where the partite sets have sizes m and n.
$d(v)$ is the degree of vertex v in G.
In a digraph $d^+(v)$ and $d^-(v)$ are the in-degree and out-degree of vertex v.

Colorings and bipartite graphs

A k**-coloring** of a graph G assigns to each vertex of G one of k colors so vertices colored the same color form an independent set. A 2-colorable graph is called **bipartite**, and the two independent sets are called **partite sets** of G. The **chromatic number** of a graph is the minimum number of colors k so that the graph can be k-colored.

A graph is bipartite if and only if it has no odd-cycles.

Walks, trails, paths, cycles, connections

A **path** is a simple graph where the vertices can be ordered so that two vertices are adjacent if and only if they are consecutive in the list. In a **cycle**, the vertices are ordered around a circle.

A **walk** in G is a list $v_0, e_1, v_1, \ldots e_k, v_k$ of vertices and edges so that each e_i has endpoints v_{i-1} and v_i. A **closed walk** has $v_0 = v_k$. A **trail** is a walk with no repeated edges. A **path** in G has no repeated vertices or edges, while a **cycle** has no repeated vertices except $v_0 = v_k$. The **length** of a walk, trail, path, or cycle is the number of edges.

Removal of a **cut-edge** or a **cut-vertex** disconnects a component. An **induced subgraph** of G is a subgraph obtained by deleting some vertices (and only the edges connected to those vertices). If the remaining vertices are T, we say it is the subgraph of G **induced by** T.

Theorem: An edge is a cut-edge if and only if it belongs to no cycle.

Theorem: Every u, v walk contains a u, v path.

Theorem: Every closed odd walk contains an odd cycle. (Good induction proof to study.)

Eulerian circuits

A graph is **Eulerian** if it has a closed trail (or **circuit**) containing all its edges. A connected graph G is Eulerian if and only if all its vertices have even degree. A connected digraph is Eulerian if and only if $d^+(v) = d^-(v)$ for all vertices v. In the proof, we showed (by induction) that every even graph decomposes into cycles and then stapled the cycles together.

For a connected nontrivial graph with exactly $2k$ odd vertices, the minimum number of trails that decompose it is k (except it's 1 if $k = 0$).

Vertex degrees and counting

$$\sum_{v \in V(G)} d(v) = 2e(G)$$

So every graph has an event number of odd degree vertices.

Trees and distance

A **forest** is an acyclic graph. A **tree** is a connected acyclic graph. A **leaf** is a vertex of degree 1. A **spanning tree** is a connected subgraph of G with vertex set $V(G)$.

Every tree with at least two vertices has at least two leaves.

For an n-vertex graph G, the following are equivalent:

1. G is connected and acyclic.
2. G is connected and has $n - 1$ edges.
3. G is acyclic and has has $n - 1$ edges.
4. For each $u, v \in V(G)$, there is exactly one u, v-path.

Every edge of a tree is a cut-edge. Adding one edge to a tree forms exactly one cycle. Every connected graph contains a spanning tree.

If T and T' are spanning trees of a connected graph G and $e \in E(T) - E(T')$, then

1. there is an edge $e' \in E(T') - E(T)$ such that $T - e + e'$ is a spanning tree of G, and
2. there is an edge $e' \in E(T') - E(T)$ such that $T' + e - e'$ is a spanning tree of G.

If G has a u, v-path, the **distance** from u to v, $d(u, v)$, is the least length u, v-path. If G has not such path, $d(u, v) = \infty$. The **diameter** is $\max_{u,v} d(u, v)$.

There are n^{n-2} trees on vertex set $\{1, \dots, n\}$. Each is uniquely determined by its **Prüfer code** $(a_1, \dots, a_{n-2})$ obtained as follows: At the i^{th} step, delete the least remaining leaf, and let a_i be the neighbor of this leaf.

Algorithms

- Kruskal's algorithm first sorts the edges by weight. Then, for each edge, include the edge in the MST if the endpoints are in differing connected components of the MST so far.
- A maximum matching on a bipartite graph can be found by repeatedly augmenting along alternating paths.
- Single-source shortest paths for non-negative edge weights can be found by Dijkstra's algorithm. Grow a tree always adding the vertex with the cheapest path from the source which extends the tree by only one edge.

Matchings and min-max theorems

Hall's Theorem states that an X, Y-bigraph G has a matching that saturates X if and only if $|N(S)| \geq |S|$ for all $S \subseteq X$.

A **vertex cover** of a graph G is a subset of vertices so that each edge has an endpoint in the cover. An **edge cover** is a subset of edges so that each vertex is adjacent to an edge in the cover. For a graph G, define:

$$\begin{aligned}
\alpha(G) &= \text{maximum size of an independent set in } G \\
\alpha'(G) &= \text{maximum size of a matching in } G \\
\beta(G) &= \text{minimum size of a vertex cover for } G \\
\beta'(G) &= \text{minimum size of an edge cover for } G
\end{aligned}$$

- (König, Egerváry) If G is bipartite, $\alpha'(G) = \beta(G)$.
- For all graphs, $\alpha(G) + \beta(G) - n(G)$, since S is an independent set if and only if $\overline{S}$ is a vertex cover.
- If G has no isolated vertices, $\alpha'(G) + \beta'(G) = n(G)$.
- If G is bipartite with no isolated vertices, $\alpha(G) = \beta'(G)$.

Planar graphs

A **planar embedding** is a drawing of a graph G in the plane without crossings. A graph is **planar** if it has a planar embedding. **Euler's formula** (which we proved by induction) states that in any planar embedding of a graph with r regions and c connected components,

$$r - c - e(G) + n(G) = 1.$$

In any connected, planar, simple graph with $n(G) \geq 3$ vertices has $e(G) \leq 3n(G) - 6$. (The proof used Euler's formula and the fact that every region was bounded by at least 3 edges.)

As a corollary, every planar, simple graph has a vertex v of degree $d(v) \leq 5$. Hence, by induction, every planar graph is 6-colorable. A slightly more complicated proof shows every planar graph is 5-colorable. It's known that every planar graph is 4-colorable.

A graph G is **homeomorphic** to H if it can be obtained by adding and deleting degree 2 vertices. Adding a vertex consists of placing along an existing edge, and removing it consists of leaving a single edge in place of the two edges leaving the vertex. **Kuratowski's theorem** states that a graph is planar if and only if it does not contain a graph homeomorphic to $K_{3,3}$ or K_5.

Flow networks and maximum flows

- Capacities satisfy $c(u, v) \geq 0$. Flows satisfy

  Capacity constraints : $0 \leq f(u, v) \leq c(u, v)$

  Flow conservation: $\forall u \in V - \{s, t\} : \sum_{u \in V} f(u, v) = 0$
- Augmenting paths consist of forward edges (edges (u, v) with $f(u, v) < c(u, v)$) and back edges (edges (u, v) with $f(v, u) > 0$).
- The min-cut max-flow theorem proves the maximum flow is equal to the minimum capacity over all cuts. Further, if there are no augmenting paths in the a flow, the flow is a maximum flow.
- The Ford-Fulkerson algorithm finds paths from s to t in the residual graph to augment the flows until no more paths can be found.