An excellent resource for Graph Theory powerpoint slides:
http://people.qc.cuny.edu/faculty/christopher.hanusa/courses/634sp11/Documents/Forms/AllItems.aspx

Notes Packet 1

Graph Definition

Figure n1.1
A graph G = (V, E, f)
V is the vertex set (e.g. w, x, y...)
E is the edge set (e.g. e_1, e_2...)
f is the function mapping edges to vertices (i.e. f: $e_1 \longrightarrow$ xw, $e_2 \longrightarrow$ xw, $e_3 \longrightarrow$ yw ...)

Simple Graph Definition

A **simple graph** G (aside from the null graph) is a graph with n $\geq$ 1 vertices and m $\geq$ 0 edges that consists of a vertex set
V(G) = $\{v_1, v_2 ... v_n\}$
and an edge set
E(G) = $\{e_1, e_2 ... e_m\}$
where each edge is an unordered pair of vertices.
An example graph theory problem: Can x workers do y jobs such that all of the jobs can be done by at least one worker?
Here's a table of workers and the corresponding jobs they can do.

Worker #	Job
1	a
2	a, b
3	b, c, d
4	a, b

Here's a graph of what that might look like:
Figure n1.3
If we look at jobs c and d, we notice that both can only be done by worker #3.
Figure n1.4

Subgraph Definition

A **subgraph** of a graph G is a graph H such that V(H) $\subset$ V(G) and E(H) $\subset$ E(G).
We write this as H $\subset$ G

Notes Packet 2

A **path** is a sequence of unique edges that connects a sequence of unique vertices. You can't repeat any vertices or edges, but the path doesn't have to contain all the edges or vertices from the graph.
A **cycle** is a path that, instead of its endpoint being another unique vertex, it ends at its beginning vertex. Both vertices and edges are not allowed to be repeated here as well.
A graph is **connected** if, given any two vertices a and b, there is a path between a and b.
A **simple** graph G is a graph such that G has no **loops** (an edge such that its two endpoints are the same vertex), and, given any two vertices a and b, a and b have no more than one edge between them.

A **loopless** graph G has all the qualities of a simple graph, but allows any two vertices to have more than one edge between them.
The **Adjacency Matrix** of a graph G, A(G), is a matrix whose entries are the number of edges between two vertices. It is symmetric if the graph is undirected.
The **Incidence Matrix** of a graph G, M(G), is a matrix whose entries indicate whether a vertex and an edge are **incident** (i.e. touching or connected). The number of rows doesn't necessarily equal the number of columns (i.e. |E(G)| = |V(G)| can be true or false).
For example, here is a loopless graph G:
Figure n2.2
And here is A(G) and M(G):
Example:

$$\begin{array}{c c} & \begin{array}{cccc} w & x & y & z \end{array} \\ \begin{array}{c} w \\ x \\ y \\ z \end{array} & \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 2 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{array} \qquad Figure - n2.2 \qquad \begin{array}{c c} & \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{array}{c} w \\ x \\ y \\ z \end{array} & \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

A(G) - Adjacency Matrix of G

M(G) - Incidence Matrix of G

The **degree** of a vertex x, d(x), is how many edges are incident (connected to) x.
A(G) and M(G) are related like so:

$$\text{A(G)} + \begin{pmatrix} d(w) & & & \\ & d(x) & & \\ & & d(y) & \\ & & & d(z) \end{pmatrix} = MM^T$$

Edges can also cross one another. From most perspectives, G_1 and G_2 are the same graph.
Figure n2.4, Figure n2.5
In fact, G_1 and G_2 are **isomorphic**.
f: V(G_1) $\longrightarrow$ V(G_2)
1 $\longrightarrow$ a
2 $\longrightarrow$ b
...
5 $\longrightarrow$ e
ij ∈ E(G_1 iff f(i)f(j) ∈ E(G_2)
The full definition is this:
An **isomorphism** from G to H is a **bijection** (a 1 to 1 correspondence) f: V(G) $\longrightarrow$ V(H) such that given an edge uv, uv ∈ E(G) iff f(u)f(v) ∈ E(H)
We say G ≅ H to indicate that G is isomorphic to H.
The **compliment** of a graph G, notated as $\bar{G}$, is a graph with the same vertex set as G, but if uv ∈ E(G), then uv ∉ E($\bar{G}$) and vise versa.
G ≅ H iff $\bar{G} \cong \bar{H}$

Bipartite Graph Definition

A **bipartite graph** G, also called a **bigraph**, is a graph st V(G) can be decomposed into two **disjoint** sets (i.e. any vertex v ∈ V(G) is either in one set or the other, not both) such that no two vertices within the

same set are **adjacent** (connected by an edge). A bipartite graph is a special case of a **k-partite** graph with k = 2.
Some other vocabularly:
A **complete** graph, K_n, is a simple graph with n vertices such that there is one edge between any two vertices.
The number of edges in K_n, $|\mathrm{E}(\mathrm{K}_n)|$, is equal to $\binom{n}{2}$. Mathematically,

$$|\mathrm{E}(\mathrm{K}_n)| = \frac{n(n-1)}{2} = \binom{n}{2}$$

$\mathbf{C}_n$ is shorthand for a cycle with n vertices.
$\mathbf{P}_n$ is shorthand for a path with n vertices and n - 1 edges.
$\mathbf{K}_{r,s}$ is shorthand for a complete, bipartite graph such that the independent sets have r and s vertices, respectively.
The notation K_5 - {e} means a complete graph with 5 vertices minus an edge e.

Decomposition

A **decomposition** of a graph G is a set of subgraphs H_1, H_2, ... H_k that partition the edges of G.
That is, ∀ i, j,

$$\bigcup_{1 \le i \le k} H_i = G, \qquad E(H_i) \cap E(H_j) = \emptyset$$

An **H-decomposition** is a decomposition of G such that each subgraph H_i in the decomposition is **isomorphic** to H.
For example, we know that K_5 - {e} decomposes into $3\mathrm{P}_4$ (3 paths with 4 vertices):
Figure 2.6
Can K_6 decompose into $5\mathrm{P}_4$?
Yes. Using the "rotation" trick.
Figure 2.7
How many copies of P_4 can K_8 decompose into?
So, K_8 has $\frac{8(8-1)}{2} = 23$ edges.
However, P_4 has 4 vertices, and therefore 3 edges.
There is no integer k such that $k\mathrm{P}_4 = 23$, so the answer is: it can't.

Peterson Graph Definition

A **Peterson graph** G is a graph with
$\mathrm{V(G)} = \{\mathrm{A} : \mathrm{A} \subset \{1, 2, 3, 4, 5\}$, and $|\mathrm{A}| = 2\}$
and
$\mathrm{AB} \in \mathrm{E}$ iff $\mathrm{A} \cap \mathrm{B} = \emptyset$
In other words, if the vertex is named 34, then it can only be connected to vertices that have neither 3 nor 4 in the name (i.e. 51, 52, 12)
Example:
Figure N2.8

Notes Packet 3

Induction Example

P(n) is true for every positive integer n.
Step **(1)**: P(1) is true (should be a trivial case).
Step **(2)**: Assume that, $\forall$ k $\geq$ 1, P(n) is true for n = k
Step **(3)**: Prove that if P(n) is true for n = k, then P(n) is true for n = k + 1 (this is "weak" induction)
Step **(4)**: This proves that since P(1) is true, P(2) is true. Since P(2) is true, P(3) is true. And so on...

A **walk** is a sequence of edges and vertices in a graph (i.e. a path with no restrictions, so you can cross over the same edge and/or vertex more than once).
A **trail** is a walk with no repeated edges. Repeated vertices are still okay.
A **path** is, again, a sequence of connected edges and vertices in a graph such that there are no repeated edges nor vertices.
A **circuit** is a cycle that allows repeated vertices. No repeated edges still, though. Also known as a **closed trail**.
A **cycle** is a path that begins and ends at the same vertex. Also known as a **closed path**.
A **u, v walk / trail / path** is a walk / trail / path that begins at u and ends at v.
In the above definition for u, v walk / trail / path, u and v would be the **endpoints**.
The **internal vertices** are the vertices that aren't endpoints.
The **length** of a walk / trail / path / circuit / cycle is how many edges it has.

Lemma 1.2.5

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

Proof.
Insert proof here.
Figure N3.1 □

Be careful, though. If you have a u, v path and a v, w path, that doesn't mean you have a u, w path containing both a u, v path and a v, w path (they might not necessarily be head to tail).

Connected / Disconnected Graphs

A graph is **connected** if there is a path between any two vertices. It is **disconnected** $\exists$ a vertex such that that's not the case.
If G is disconnected, then we can think of G as a set of components.
A component is **non-trivial** if it has at least one edge.
Adding an edge can either decrease the number of components, c(G), or not change it.
Deleting an edge can either increase c(G) or not change it.
Deleting a vertex (which also deletes the attached edges) can increase c(G) by n where $0 \leq n \leq \infty$.

Proposition 1.2.11

If you have a graph with n vertices and k edges, then c(G) $\geq$ n - k
An edge is **cut-edge** if removing the edge increases c(G) (i.e. the edge is a "bridge").
A vertex is a **cut-vertex** if removing the vertex increases c(G).
An **induced subgraph** of G is a subgraph $H \subset G$ such that if $e \in E(G)$ and e's endpoints are both members of V(H), then $e \in E(H)$. (i.e. G[T] where $T \subset V$)

Theorem 1.2.14

e is a cut-edge iff e belongs to no cycle
INSERT FULL PROOF HERE

Proof.
$\longrightarrow$
$\longleftarrow$ By contrapositive: e is not a cut-edge $\longrightarrow$ e belongs to a cycle □

Recall: Bipartite graphs
Figure n3.2

Theorem 1.2.17

A graph G is bipartite iff there are no odd cycles.
INSERT FULL PROOF

Proof.
$\longrightarrow$
$\longleftarrow$ If there is no odd cycle then every walk is odd. Any closed walk can be decomposed into cycles. □

To prove G is bipartite, find a bipartition X and Y.
To prove G is not bipartite, find an odd cycle.
To test if G is bipartite or not uses an algorithm of O(|E[G]|)

Eulerian Graph Definition

A **Eulerian** graph G is a graph containing a Eulerian circuit. It can also mean a graph such that every vertex is of even degree. These definitions coincide for connected graphs.
A **Eulerian Circuit** is a circuit containing all the edges of a graph.
In other words, if a graph is connected and every vertex has even degree, then it has at least one Eulerian Circuit.
If a graph has more than 2 vertices of odd degree, then it cannot have a Eulerian Circuit or Eulerian Path.
If a graph is connected and has exactly 2 vertices of odd degree, then it has at least one Eulerian Circuit.
An **even** graph G is a graph such that $\forall$ v $\in$ V(G), deg(v) is even.
Lemma: If $\forall$ v $\in$ V(G), deg(v) $\geq$ 2, then G contains a cycle.
The **maximum** path is the path with the largest number of edges (if there's a tie there is no maximum path).
A **maximal** path is a path such that it is contained by no larger path.

Theorem 1.2.26

A graph G is Eulerian iff c(G) $\leq$ 1 and G is an even graph $\longrightarrow$
$\longleftarrow$
Induction on |E| = m

Application 2.3.10 - The Chinese Postman Problem

Pictures, maybe?
Something about weighted K_8...
Something about the Hungarian theorem/algorithm
Find the minimum total weight of 4 edges that pair up the 8 vertices.
2nd to last page of packet 3, there's some pictures that maybe go here

Proof Technique: Extremality

Proposition 1.2.28: if, $\forall$ v $\in$ V(G), d(v) $\geq$ k, then $\exists$ a path of length k.
There's some other stuff, but some is repeats (maximum, maximal...)

———————————————Side Note———————————————-

END OF NOTES FROM BEFORE TEST 1, BEGINNING OF POST TEST 1 NOTES

Definition of a king:
For any x, either King $\longrightarrow$ x, or King $\longrightarrow$ y $\longrightarrow$ x for some path y.
Prop 1.4.30 - Every tournament has a king
—–
A graph is **acyclic** if it has no cycle.
A graph is a **forest** if it is acyclic.
A graph is a **tree** if it is a connected acyclic graph.
¡pictures of trees¿
A **leaf** is a pendant vertex (i.e. a vertex with degree 1)
A **star** is ***
¡picture of a star¿
The **distance**, d(u, v), is the length of the shortest path between two vertices u and v.

Lemma 2.1.3:

Every tree G st $|V(G)| \geq 2$ has ≥ 2 leaves.
Deleting a leaf results in a smaller tree on n - 1 vertices.

Proof.

¡picture of maximal path. i.e. dot-dot-dot-dot¿
No leaf is an internal vertex of a path.
We would use an induction method to prove this:
B *** A(n) $\Rightarrow$ B(n)
A(n): T is a tree on n vertices
B(n): T has n - 1 edges
Want to show: num edges = num vertices - 1
¡picture from top right of Method of Induction page¿
Induction on n:
Step 1:
T′ = T - {a leaf}
T′is a tree on n - 1 vertex
Step 2:
T′has n - 2 edges (induction hypothesis)
Step 3:
T = T′+ {an edge}
T has n - 2 + 1 = n - 1 edges.

□

Theorem 2.1.A (or 4?)

a. connected, no cycle. n vertices (do I have n - 1 edges?)

b. connected, n - 1 edges

c. n - 1 edges, no cycle (not sure if connected)

d. For any u, v ∈ V, ∃ exactly one u, v - path. No loops.

Proof.

We're going to say these three things are equivalent.
We did A ⇒ B in previous slides. (induction on n)
For B ⇒ C:
Want to show: G has no cycles
Suppose G has cycles (contradiction):
¡picture from Theorem 2.1.A (or 4)
G′ = G - {e$_1$, e$_2$, ...} is acyclic
acyclic, connected, n - 1 vertices = tree
G′is connected, (using any tree that has n vertices has n - 1 edges), G′has n - 1 edges
C ⇒ A (if you have 3 and 2, then prove you have 1):
Suppose c(G) (number of components) = k (by contradiction).
¡pictures of n$_1$ vertices, n$_2$ vertices.. n$_k$ vertices¿ ¡has n$_1$ - 1 edges, n$_2$ - 1 edges, etc...¿

$$n - 1 = e(G) = \sum_{i=1}^{k}(n_1 - 1) = \sum_{i=1}^{k}(n_i - k) = n - k$$

The only solution is that k = 1.

□

Corollary 2.1.5

a. Every edge of a tree is a cut-edge.

b. Adding one edge to a tree forms exactly one cycle.

c. Every connected graph contains a spanning tree.

A spanning subgraph of G is a subgraph of G that contains all the vertices of G.
A spanning tree is a spanning subgraph that is a tree.

Proposition 2.1.8 (or B)

Tree T has k edges, simple graph G has min(G) ≥ k (minimum degree bigger than or equal to k) ⟶ T is a subgraph of G.
T′ = T - {a leaf} has k - 1 edges.
¡picture of G¿
To prove this, we would use induction on k.
min vertex(G) ≥ k ≥ k - 1
T′has k vertices.
Base: k = 1
If T has only 2 vertices, then T has 1 edge. This is a trivial case.
¡Missing some other stuff¿

Definition 2.1.9

eccentricity (for any connected graph) ϵ (u) = max{d(u, v) : v $\in$ V(G)}
¡picture below eccentricity¿ (where 4 is the radius, 7 is the diameter)
The radius, rad(G), is the minimum *** = min of ϵ (u) where u $\in$ V
The diameter, diam(G), is the maximum *** = max of ϵ (u) where u $\in$ V
ϵ (u) = d(u, v) for some leaf v

Theorem 2.1.13 (Jordan, 1869)

The center of a tree is always one edge or one vertex.

Proof.

We do induction on n.
Let: T$'$ = T - {all leaves}
ϵ $_{T'}$(u) = ϵ $_T$(u) - 1
If G $\neq$ a line segment with a vertex at each end, then no leaf can be a center vertex.

□

Theorem 2.1.10 [Not On Test]

Not on test

Theorem 2.1.11

G is simple, Diam(G) $\geq$ 3 $\longrightarrow$ Diam($\overline{G}$) $\leq$ 3

Proof.

Claim: x cannot be adjacent to both u and v, otherwise distance will be smaller than 3. So, at least one of them is not true.
Dotted lines signify non-adjacency.
In the case of neither x nor y being adjacent to u, then in $\overline{G}$, u is adjacent to both x and y.
In the case

□

hi

How many simple graphs with vertex set [n] are there?
[n] = {1, 2, 3, ... n}
In other words, how many labelled graphs on n vertices are there?
Answer: $2^{\binom{n}{2}}$
How many trees with vertex set [n] are there?
Cayley's formula: n^{n-2} (proof is very complicated, only need to understand conclusion) (the number of labelled trees on n vertices)
Labelled trees: 1:1 (a$_1$, a$_2$, ... a$_{n-2}$) : a$_i$ $\in$ [n]
Suppose I have 3 vertices. How many labeled trees can I get?
Answer: 3 possible labeled trees. If you have vertices 1, 2, and 3, then you can have 3 graphs: **(1)** 12, 23, **(2)** 23, 31, **(3)** 31, 12
Let's say I have a complete graph, K$_n$, with labelled vertices. How many labelled spanning trees does it have? Answer: n^{n-2} (Cayley's formula)

To generalize the problem:

Contraction of edge:

¡picture with edge uv and vertices A, B, C $\longrightarrow$ picture with multiple edges between w and each vertex in C¿

Example 2.2.9:

¡picture of a square with a diagonal edge, same picture minus diagonal edge (G - e), bird (G dot e))

Proposition 2.2.8

τ(G) (the number of spanning trees of G) = τ(G - e) + τ(G dot e)

Let: T be a spanning tree of G

case i: e (the diagonal edge in the picture) is not in E(T)

Any spanning tree of the original graph without using the diagonal is still a spanning tree. If you don't use the diagonal edge, it's from G - e.

case ii: e (the diagonal edge in the picture) is in E(T)

You should prove Prop 2.2.8 for practice.

Remark 2.2.10 (Basis case for computing τ(G))

Suppose G has no cycle other than multiple edges. Then:

τ(G) = {product of edge multiplicities if G is connected, 0 if disconnected}

¡Remark picture¿

1 * 2 * 3 choices

You're encouraged to try an example. For example: K$_4$

For a general graph, the Cayley formula doesn't work. Here's a 3rd way:

Theorem 2.2.12

Let: G be a loopless n-graph (graph with n vertices)

Q = (q$_{ij}$)$_{n\times n}$ is defined by:

q$_{ij}$ = {d(v$_i$) if i = j, $-a_{ij}$ if i $\neq$ j}

a$_{ij}$: the number of edges joining v$_i$ and v$_j$

Q* : obtained by deleting any row s and column t of Q.

Then τ(G) = $(-1)^{s+t}$ det Q*

¡picture 3¿

The sum of every row and every column is equal to 0. Why?

The diagonal talks about the degree, but the off diagonal takes off each edge.

Determining the determinant of the matrix Q doesn't give you any new information, but deleting any row or column and making a submatrix, Q$_*$, of the matrix, and then taking the determinant will.

The absolute value of det Q$_*$ (or just multiplying by $(-1)^{s+t}$ will give you τ(G)

(we took both a row (1st row) and a column (1st column) to make it symmetric and make it easier to take the determinant)

Conjecture 2.2.13 (still open)

K$_{2m+1}$ decomposes into 2m + 1 copies of T with m edges.

So K$_{2m+1}$ has $\frac{(2m+1)(2m)}{2}$ edges

Therefore, there are 2m + 1 copies of T.

Graceful labeling: a 1-1 correspondence between the vertices and a set of numbers for each vertex (packed very closely from 0 to n - 1 (where n is the number of vertices))

f: V $\longrightarrow$ distinct number $\in$ {0, 1, ... m} (a bijection from a vertex set to "this" set)

For example. If T has m edges, how many vertices do you have? Answer: m + 1

You want every vertex to have a distinct number.

¡graceful labelling picture¿

f(uv): = |f(u) - f(v)|

{f(u, v): uv $\in$ E} = {1, 2, ... m}

Conjecture 2.2.15: The Graceful Tree Conjecture (Kotzig, Ringel, 1964)

Open, stronger than conjecture 2.2.13.
Every tree has a graceful labeling.

Theorem 2.2.16

T is graceful $\Rightarrow$ Conjecture 2.2.13 holds
¡picture 1, picture 2 - examples of this¿
Remember, we want all edges to receive different numbers (Graceful labelling), so when we do a rotation, all of the different edges are covered exactly once
displacement(i, j): number of unit moves from i to j
K$_{2m+1}$
end of 10/11 lecture
beginning of 10/16 lecture

Caterpillar (special type of tree)

Example of a caterpillar tree:
Figure n6.1
Any caterpillar is graceful.
Figure n6.2

Theorem 2.2.19

T is a caterpillar iff T has no component like this:
Figure n6.3

Proof.

T$'$= T - {all leaves}
What is T$'$?
A path.
T is a caterpillar iff T$'$is a path
T has no Figure n6.3 iff T$'$has no Figure n6.4
NEED MORE PROOF

□

Minimum Spanning Tree Problem

A super important problem in Graph Theory.
Suppose we have a really remote area. There are a few villages, with some distances between all of them. We know all of the distance between any two villages.
A spanning tree is the most efficient structure in the sense that it has the fewest (number of edges) while still being connected.
A **weighted graph** G is a graph s.t all the edges of G have nonnegative weights. In other words, for a graph G, there is a function W that maps all e $\in$ E(G) to nonnegative weights.
If G is the following graph:
Figure n6.5

What is the minimum weight of a path from x to y?
The running time to solve this is O(f(n), where
the number of computational steps ≤ c|f(n)| for n ≥ a
where c is a constant and n is the size of the input.
The **minimum spanning tree** is the tree of a graph that connects all the vertices of a graph with edges such that

$$\sum_{e \in T} W(e)$$

is minimized.

Example 2.3.2 - The Greedy Algorithm

Let: G be the below graph
Figure n6.6
Algorithm 2.3.3:
Set V(H) = V(G), E(H) = ∅
If H is disconnected, add to H the cheapest edge (the edge with the smallest weight) joining two components of H.