MATC32: Graph Theory
Lecture Notes

Joshua Concon

University of Toronto Scarborough – Fall 2017

Pre-reqs are MATB24, which is the second course on linear algebra at UTSC. Instructor is Dr. Louis de Thanhoffer de Volcsey. I highly recommend sitting at the front since he likes to teach with the board. If you find any problems in these notes, feel free to contact me at conconjoshua@gmail.com.

Contents

1 Tuesday, September 5, 2017 **4**
- 1.1 The Seven Bridges of Konigsberg . . . 4
- 1.2 (Outline) Solution to Konigsberg . . . 6

2 Friday, September 8, 2017 **7**
- 2.1 Graphs . . . 7
- 2.2 Graph Theory Applications . . . 8
  - 2.2.1 Uber . . . 8
  - 2.2.2 Monge's Theorem (on matching) . . . 8
  - 2.2.3 Marriage (Stable) Problem . . . 9
  - 2.2.4 Scheduling (Graph Colouring) . . . 9
  - 2.2.5 Network Problems . . . 10
  - 2.2.6 Uber (Pathfinding) . . . 10
- 2.3 Morphism . . . 10

3 Tuesday, September 12, 2017 **12**
- 3.1 More Morphism . . . 12
- 3.2 Decomposing Graphs . . . 13

3.2.1 Connectivity of G 15

4 Friday, September 15, 2017 **16**
4.1 Equivalence Relations on a set 16

5 Tuesday, September 19, 2017 **19**
5.1 Konig's Theorem (continued) 19
5.2 Networks 20

6 Tuesday, September 26, 2017 **22**
6.1 Flow on a network 22

7 Friday, September 29, 2017 **24**
7.1 Max-flow Min-Cut Theorem (continued) 24
7.2 (Ford-Fulkerson) Algorithm for finding a maximal flow for a network 27

8 Tuesday, October 3, 2017 **28**
8.1 Applying Min-cut Max-flow 28

9 Friday, October 6, 2017 **31**
9.1 Hall's Theorem on Perfect Matching 31
9.2 Connectivity: Menger's Theorem 32

10 Tuesday, October 17, 2017 **34**
10.1 Planar Graphs 34

11 Tuesday, October 24, 2017 **35**
11.1 Closed sets 35

12 Friday, October 27, 2017 **36**
12.1 More Definitions for Planar Graphs 36
12.2 Planar Graphs 38
12.2.1 Application of Euler's formula for connected, planar graphs 38

13 Tuesday, October 31, 2017 **39**
13.1 Subdivisions 39
13.2 Dual Graphs 40

13.3 Platonic Solids 41

14 Friday, November 3, 2017 **42**
14.1 Classifying Platonic Solids 42
14.2 Coloring 43
14.3 Bounds for colouring 44
14.3.1 Lower Bound 44
14.3.2 Upper Bound 45

15 Tuesday, November 7, 2017 **46**
15.1 Colouring Problems 46
15.1.1 Lower Bound 46

16 Tuesday, November 14, 2017 **48**
16.1 5-Colour Theorem 48

17 Friday, November 17, 2017 **51**
17.1 Graph Colouring 51
17.2 Spanning Trees 52
17.2.1 Finding a Minimum Spanning Tree 53

18 Tuesday, November 21, 2017 **55**
18.1 Kruskal's Algorithm 55
18.2 Prim's Algorithm 55
18.3 Dijkstra's Algorithm 55

19 Friday, November 24, 2017 **56**
19.1 Spanning Cycles – Eulerian Cycles 56

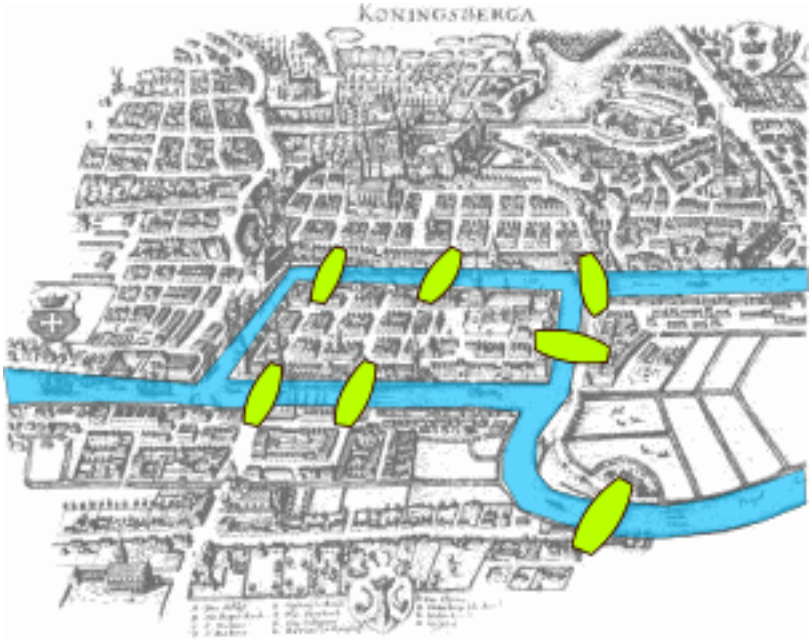
20 Tuesday, November 28, 2017 **58**
20.1 Hamiltonian Paths /Cycles 58
20.1.1 Necessary Conditions 58
20.1.2 Sufficient Conditions 58

21 Friday, December 1, 2017 **59**
21.1 Hamiltonian Graphs 59
21.1.1 Necessary Condition 59
21.1.2 Sufficient Condition 59

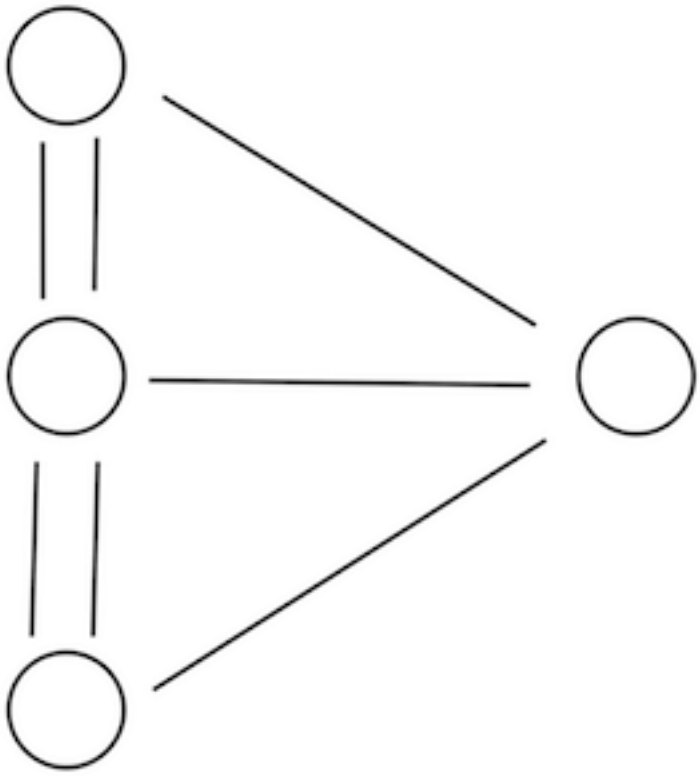
1 Tuesday, September 5, 2017

1.1 The Seven Bridges of Konigsberg

So basically there's this city called Konigsberg where a river flows through, and because of that, there are 7 bridges in the city. The bridges look like this:



There problem that came up is whether or not it was possible to walk through every bridge in the city once in the same walk. This problem was eventually solved by Euler.



It can be simplified to this. This is called a 'Graph', the circles are called 'nodes' or 'vertices' and the lines are called 'edges'. The different parts of the city are represented by the nodes and the bridges are represented by the edges.

Definition: Graph (G)

1. Contains a set $V(G)$ = the set of nodes
2. Contains a set $E(G)$ = the set of edges

A graph is called **Simple** if the graph has no loops and does not have multiple edges (i.e. Each edge is an unordered pair of distinct vertices). A graph is called a **Loop** if there is an edge that connects a vertex to itself.

Definition: Path

A set of edges denoted by vertices $v_1, v_2, ..., v_n$ where there is a node between every edge between v_i and v_{i+1} $\forall i, 1 \leq i \leq n-1$

1.2 (Outline) Solution to Konigsberg

Assume the graph has a path containing all edges $u_1, ..., u_n$.

Consider a vertex that isn't the first or last vertex travelled in the path (i.e. any vertex excluding u_1 and u_n).

There must be an even number of edges for each of the nodes in between the edges in the path (excluding the first and the last node visited, unless the first and the last node visited are the same node).

Since there are an odd number of adjacent nodes for all 4 nodes, this path does not exist. Therefore, there is no solution to Konigsberg.

2 Friday, September 8, 2017

2.1 Graphs

Definition: Graphs

A graph G consists of 2 (finite) sets:

- $V(G)$: vertex set
- $E(G)$: edge set

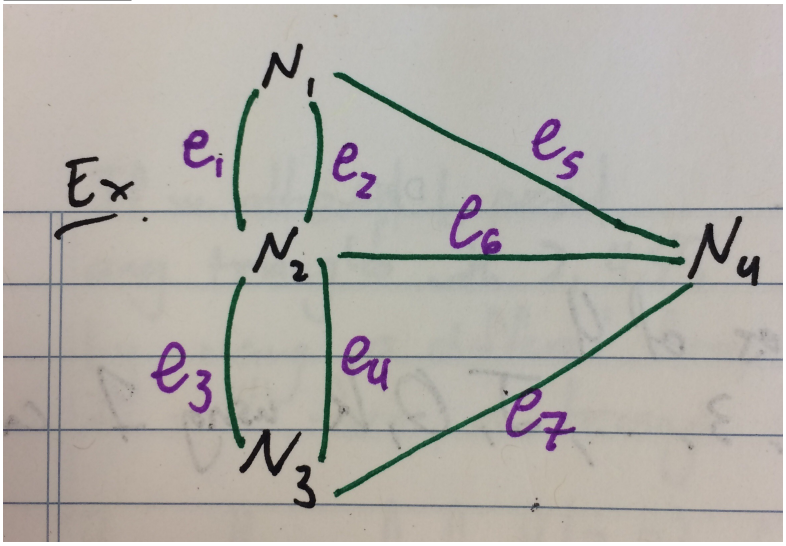
Together with an assignment from $E(G)$ to the set of subsets $V(G)$, where the subset is of size 1 or 2, containing the node(s) at the ends of the endpoints of each edge.

If an edge has the same node at both of it's endpoints, it is called a **loop**.

If 2 vertices are endpoints of more than one edge, we say that they are **multiply-edged**.

A graph without multiply-edged vertices is called **simple**.

Example:



$$E(G) = \{e_1, ..., e_7\}$$
$$V(G) = \{N_1, ..., N_4\}$$

some of the assignments of $E(G) \mapsto V(G)$ include:

$$e_5 \mapsto \{N_1, N_4\}$$

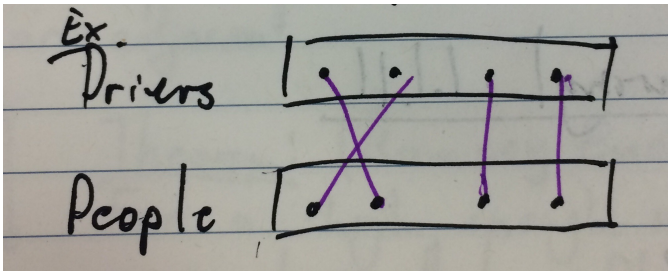
Adjacent edges are edges with a vertex that is a common endpoint.

2.2 Graph Theory Applications

2.2.1 Uber

V = People Using Uber (both drivers and passengers)

E = If it is realistic for a driver to pick up a person



Definition: Matching

A **matching** in a group is a set of edges, none of which are adjacent

Note: $V(G) = S_1 \sqcup S_2$ and there are no edges between of S_1 with respect to S_2 ($\sqcup$: refers to a union between two disjoint sets).

These two sets S_1, S_2 are independent sets

A **bipartite** graph has $V(G) = S_1 \sqcup S_2$ where both S_1, S_2 are independent.

2.2.2 Monge's Theorem (on matching)

Split a deck of 52 cards into 13 piles of 4, is it always possible to count an ace,2,3,...., Jack, Queen, King using 1 card drawn from each pile?

2.2.3 Marriage (Stable) Problem

Matching n men with n women

2.2.4 Scheduling (Graph Colouring)

Let V represent different courses, and edges between courses mean that a student can take both courses at the same time. The problem is to reduce the edges between vertices with the same label or colour.

$X(G)$ is the chromatic number of a graph G and is the minimum number of colours needed to colour a graph without the edges having endpoints with two vertices of the same colour. To distribute these colours, there is a mapping from $V(G) \mapsto C$ where C is a set of colours.

Statement: In a room of people, we can always be certain that 3 people either know each other or are all strangers for a room of 6+ people. So if we represent the people in the room as vertices in a non simple graph, and edge connections between people as the two unique people at the endpoints refer to familiarity or unfamiliarity, then you can always form a triangle with the edges.

This is the proof of $R(3,3) = 6$ for Ramsey's Theorem.

Definition: Complete Graph

A Complete Graph is a simple graph where any 2 vertices are adjacent.

A **clique** is a subset of vertices C such that all vertices are adjacent.

Theorem: Ramsey's theorem

For a high enough number $R(n,m)$, we can guarantee that after colouring a complete graph of m vertices with 2 colours (in any way), there will be a clique of n vertices of the same colour.

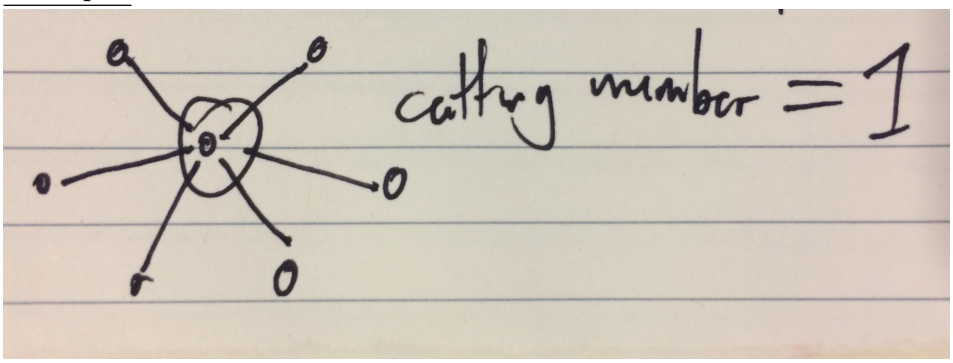
2.2.5 Network Problems

A path is a series of vertices $v_1, .., v_n$ where v_i and v_{i+1} are adjacent for all $i, 1 \leq i \leq n-1$.

Say 2 vertices are **connected** if there exists a path between them.

Cutting number of 2 vertices m, n are the amount of vertices that need to be removed such that vertices m, n are not connected.

Example:



2.2.6 Uber (Pathfinding)

Pathfinding is a graph problem where V represents all the intersections of a map and E is all the roads. For a path $v_1, ..., v_n$, the length is the number m, which is the sum of all the edge weights ≥ 0, and for 2 connected vertices, we are trying to find the path of the least length. Dijsktra's Algorithm solves this problem.

2.3 Morphism

A morphism of graphs $G \mapsto G'$ consists of 2 functions:

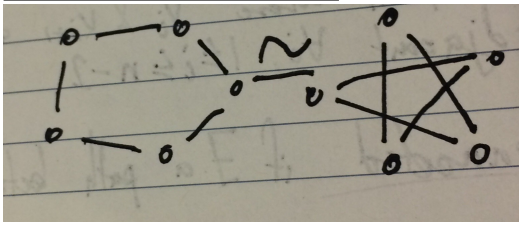
- $\gamma_1 : V(G) \mapsto V(G')$
- $\gamma_2 : E(G) \mapsto E(G')$

$\forall e$, if $e \in E(G)$ has endpoints v_1, v_2, this implies that $\gamma_2(e)$ has endpoints $\gamma_1(v_1), \gamma_1(v_2)$.

Note: This definition is more general than the book.

An isomorphism is a morphism such that γ_1, γ_2 are both bijective.

Example of a morphism: (Note that $\simeq$ denotes a morphism)



3 Tuesday, September 12, 2017

3.1 More Morphism

Observation 1

if G' is simple, there exists a morphism $G \mapsto G'$ and so $\gamma_1 : V(G) \mapsto V(G')$ such that if $e \in E(G)$ has endpoints v_1, v_2, then $\gamma_1(v_1), \gamma_1(v_2)$ are adjacent.

A **subgraph** of G is a subset $V' \subset V$, $E' \subset E$ such that $Id : (V', E') \mapsto G$ is a morphism.

An **isomorphism** is when a morphism $G \mapsto G'$ is bijective (That there is a one to one relationship between $E \mapsto E'$ and $V \mapsto V'$)

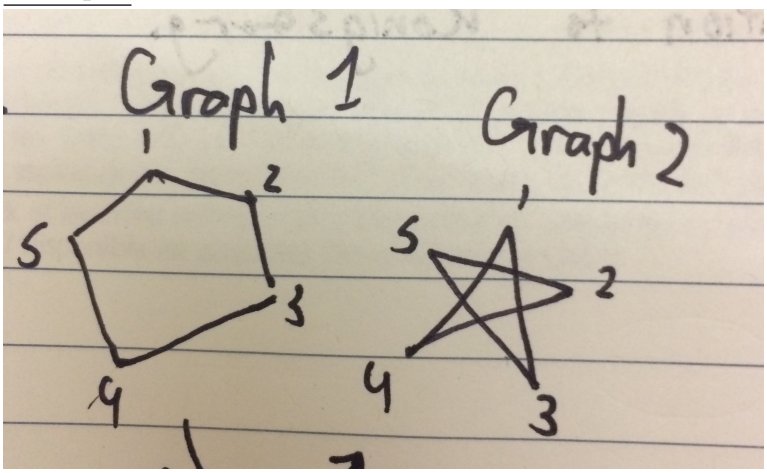
Exercise: if $(\gamma_1, \gamma_2) : G \mapsto G'$ is an isomorphism, then $(\gamma_1^{-1}, \gamma_2^{-1}) : G' \mapsto G$

Recall: Bijective implies injective and surjective.

Definition: Complement of Simple Graphs

The complement of a simple graph G is $\overline{\overline{G}} = (\overline{\overline{V}}, \overline{\overline{E}})$ where $V(\overline{\overline{G}}) = V(G)$ and $\overline{\overline{E}}$ does not have any edge in E, but if $v_1, v_2 \in V$ are not adjacent, then v_1, v_2 are adjacent in $\overline{\overline{G}}$ for all $v_1, v_1 \in V$

Example:



Graph 1 and Graph 2 are complements of each other.
We say a graph is **self-complementary** if a graph is isomorphic to its complement.

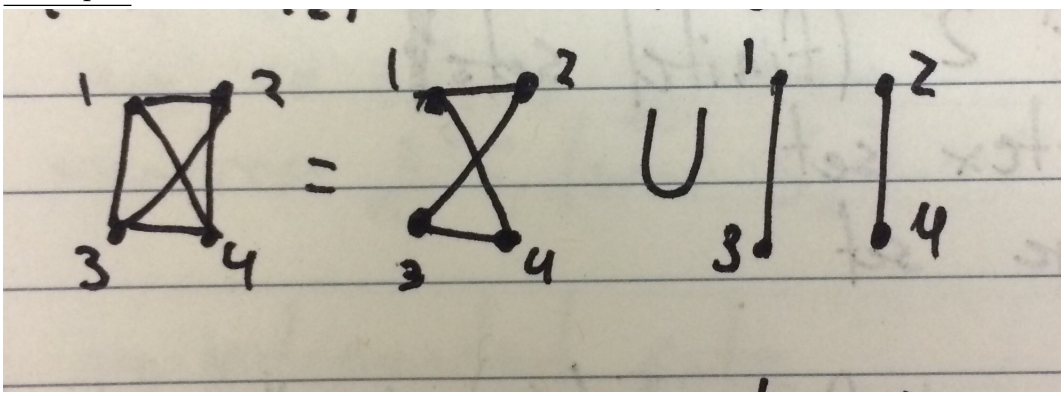
3.2 Decomposing Graphs

Definition: Union of Graphs

if G_1, G_2 are two graphs, assume $V(G_1), V(G_2) \subset V$ and $E(G_1), E(G_2) \subset E$. Then for a Graph $G_1 \cup G_2$

- $V(G_1 \cup G_2) = V(G_1) \cup V(G_2)$
- $E(G_1 \cup G_2) = E(G_1) \cup E(G_2)$

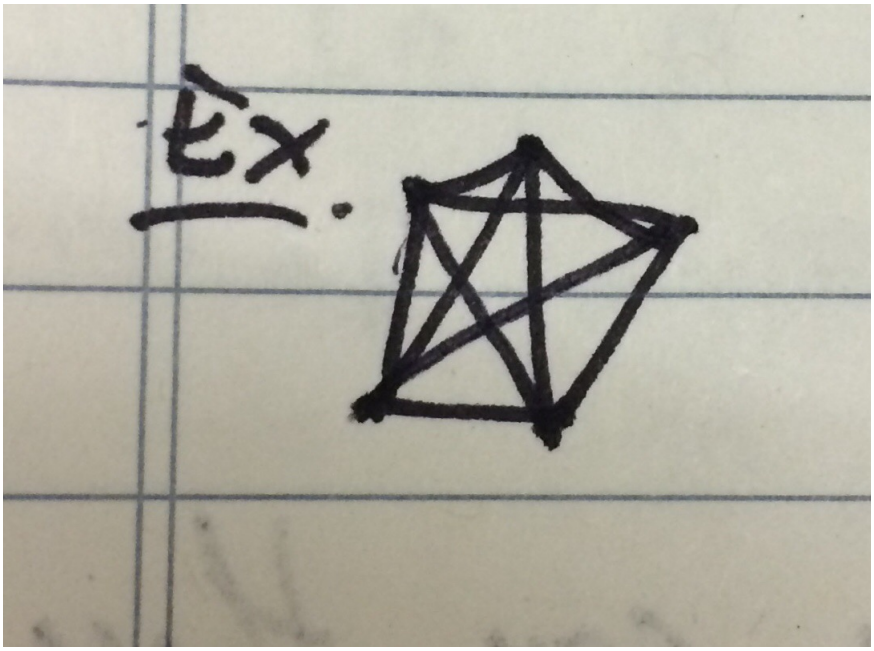
Example:



We say G **decomposes** into G_1 and G_2 if the following:

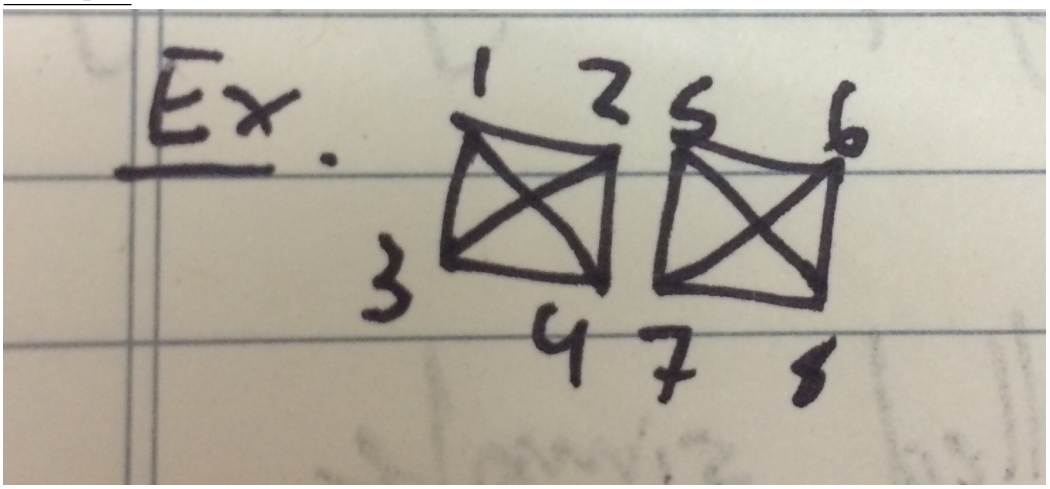
- $G_1 \cup G_2 = G$
- if $\forall e \in E(G)$ there exists one i such that $e \in E(G_i)$ (so if the edges between G_1 and G_2 don't overlap)

Example:



This is k_5, the complete simple graph of 5 vertices (complete meaning all vertices are adjacent)

Example:



This is the disjoint union of 2 graphs. $G_1 \sqcup G_2$ is the union of 2 graphs with $V(G_1) \cap V(G_2) = \emptyset$

3.2.1 Connectivity of G

Path A path is a simple subgraph such that we can order vertices $v_1, ..., v_{n-1}$ in such a way that exactly v_i, v_{i+1} are adjacent $\forall i$

Cycle A cycle is a subgraph that is simple and that we the vertices $v_1, ..., v_n$, $v_1 = v_n$ such that exactly v_i, v_{i+1} are adjacent $\forall i$ and that all edges are unique.

Note: the arrangement of vertices in a path is called a **walk**, the arrangement of vertices in a cycle is called a **trail**.

we refer to an edge by (u, v) where u, v are the start and end vertices.

4 Friday, September 15, 2017

4.1 Equivalence Relations on a set

i.e. Let X be the set of people in a room and the pair $(p_1, p_2) \in R$ if $p_1 \in X$ and $p_2 \in X$ are the same age.

An equivalence relation can be thought of as a certain property that is equal between 2+ elements.

Note: For R which is the set of people in X that have the same age, you can subdivide X into disjoint subsets of R. We can rename R as x_i where i is the age of the people in the set (a partitioned x)

$$\bigcup x_i = X$$

$$x_i \cap x_j = \emptyset, i \neq j$$

Definition: Equivalence Relation

A set of pairs R is an equivalence relation if it has the following properties:

- **Reflexive** : $\forall x \in X, (x, x) \in R$
- **Symmetric** : $\forall x, y \in X, (y, x) \in R$ implies $(y, x) \in R$
- **Transitive** : $\forall x, y, z \in X, (x, y), (y, z) \in R$ implies $(x, z) \in R$

Some examples include people who like the same colour, have the same age, have the same major,....

Claim: If R is an equivalence relation, then $X = \{y : (x, y) \in R\}$ forms a partition.

Proof. For notation, we will write $x \sim y$
We need to prove 2 things.

1. $\bigcup_{x\in X} \overline{\overline{x}} = X$, so we pick $x \in X$ and since $x \sim x$ then that implies that $x \in \overline{\overline{x}}$

2. Union is disjoint. So we must prove that

$$(a)\overline{\overline{x}} \cap \overline{\overline{y}} = \emptyset \text{ if } \overline{\overline{x}} \neq \overline{\overline{y}} \text{ iff } (b)\exists z : z \in \overline{\overline{x}} \cup \overline{\overline{y}} \rightarrow \overline{\overline{x}} = \overline{\overline{y}}$$

   If we assume (b) is true then:

$$\begin{aligned} &\rightarrow x \sim z, y \sim z \\ &\rightarrow x \sim z, z \sim y \text{ by symmetry} \\ &\rightarrow x \sim y \text{ by transitivity} \\ &\rightarrow y \in x \end{aligned}$$

   next we assume $a \in \overline{\overline{x}} \rightarrow x \sim a$, but $y \sim x \rightarrow y \sim a \rightarrow a \in y$ Therefore we have that $\overline{\overline{x}} \subset \overline{\overline{y}}$ and $\overline{\overline{y}} \subset \overline{\overline{x}}$

■

SideNote: if we start with a partition of X into subsets $(x) \in l$ then the relation $x \sim y \leftrightarrow x, y$ lie in the same X is an equivalence relation with partition $(x_i)_{i\in l}$

Example: Take the set of all graphs, say G, G' and $\exists$ an isomorphism $G \mapsto G'$ and $G' \mapsto G''$ then $\exists$ an ismorphism $g \circ f : G \mapsto G''$

Reminder: graph G with vertices u, v
(u, v) = path in a simple graph where we can order vertices $u, ..., v$ such that only adjacent edges have consecutive endpoints (ordering is a walk)

For any graph G, say $x, y \in V(G)$

$$x \sim y \leftrightarrow \begin{cases} y = x \\ y \text{ is connected to } x \end{cases}$$

is an equivalence relation.

For a vertex v, the set $\overline{\overline{v}} = \{y : y \text{ connected is the connected component of } x \text{ to } v\}$

Note: The subgraph $\overline{\overline{x}}$ is connected

If $u_1, u_2 \in \overline{\overline{v}}$ which imples that $u_1 \sim v, u_2 \sim v$ which implies that $u_1 \sim v, v \sim u_2$ which implies that $u_1 \sim u_2$

Result: Every graph is the disjoint union of common subgraphs, this result characterizes edge cuts and characterizes bipartite graphs

Definition: Edge Cuts

In a graph G, an edge creates a cut if the number of connected components of $G \setminus \{e\}$ increases

If an edge lies in a cycle, it is not a cut edge or, a cut edge does not lie in a cycle.

Thereom

An edge is a cut edge iff that edge does not lie on the cycle.

Proof. Take an edge e, we need to show that $G \setminus \{e\}$ remains connected (which implies that e lies in a cycle).
Assume $G \setminus \{e\}$ is connected, let x, y be the endpoints of e, since $G \setminus \{e\}$ is connected, there is a path between $x, y \in G \setminus \{e\}$.
Now assume e lies in a cycle in G, we need to show that $G \setminus \{e\}$, or that there exists a path between nodes u, v where u is an adjacent node to x but is not equal to y and v is an adjacent node to y but is not equal to x.

Now since $G \setminus \{e\}$ is connected, there is a path between nodes u and v that does not go through edge e and a path that goes through edge e (namely, the path u, x, y, v). Therefore e is not an edge cut iff it lies on a cycle. This implies that the negation of this is true, that an edge is an edge cut iff it doesn't lie on a cycle. ■

5 Tuesday, September 19, 2017

5.1 Konig's Theorem (continued)

Recall: Can we see when a given graph is bipartite? (A graph G is bipartite if $V(G) = X \sqcup Y$ and X, Y are independent sets)

Konig's Theorem states that G is bipartite iff there are no off length cycles.

Lemma: In a graph G, any odd length walk that is also closed contains a cycle.

From this we observe that we can assume that G is connected.

TONCAS (The Obvious Necessary Conditions Are Also Sufficient): A bipartite graph cannot have odd length cycles.

Observe that a cycle must travel through an even number of edges to return to the original vertex.

Now consider the converse of this statement: If the cycle travels odd edges, the cycle will be on the other side of the partition (so if $V(G) = X \sqcup Y$ and the cycle started at X, travelling odd edges will mean the cycle will finish in partition Y.)

Let $v \in V(G)$
$X = \{u \in V(G) :$ the path of minimal length between v, u is even for all $v \in V(G)\ \}$
$Y = \{u \in V(G) :$ the path of minimal length between v, u is odd for all $v \in V(G)\ \}$

So we know from this that.

1. $G = X \sqcup Y$
2. X, Y are independent

Assume X is not independent, so let c be an edge between $u, u' \in X$, so take a path

$$p : v \to u \text{ of minimal even length}$$

and a path

$$p' : u' \to v \text{ of minimal even length}$$

now consider the path

$$v \xrightarrow{p} u \xrightarrow{c} u' \xrightarrow{p'} v$$

This is a closed walk of odd length, therefore there exists a closed cycle of odd length.
Therefore there exists a closed walk of odd length iff X, Y are not independent (i.e. a Bipartite graph).

5.2 Networks

From this we will look at the Max Flow and Min Cut problem. This involves looking at directed graphs where all the edges in the graph are assigned a flow, which can represent the carrying capacity of trains along a route. This may be important as we may need to know what would be the best way to disrupt the network.

Definition: Directed Graph

A graph G consists a set of vertices ($V(G)$) and a set of edges ($E(G)$) and

$$E(G) \mapsto V(G) \times V(G) : e \mapsto (h(e), t(e))$$

where $h(e), t(e)$ refer to the head and tail of edge e respectively.

Definition: Network

A type of directed graph with a source node s that no edge goes into and a sink node t that no edge goes out of. There is also a capacity function

$$c : E(G) \mapsto \mathbb{R} \geq 0$$

that assigns a capacity of flow for each edge in $E(G)$

Remark: For this week, assume G is simple directed graph, i.e. between any choice of $v_1, v_2 \in V(G)$, there exists at most 1 edge with a head of v_1 and a tail of v_2. We will also introduce some notation that if there exists an edge from v_1 to v_2 we denote it as

$$v_1 \to v_2$$

Definition: Capacity function

a capacity function $c : V \times V \mapsto \mathbb{R} \geq 0$ (takes in 2 vertices and returns a positive real number) and is defined by:

$$c(v_1, v_2) = \begin{cases} c(v_1 \to v_2) & \text{if } (v_1 \to v_2) \in E \\ 0 & \text{if } (v_1 \to v_2) \notin E \end{cases}$$

Definition: Flow f

a flow function is defined as $f : E(G) \mapsto \mathbb{R} \geq 0$ with the following properties:

- $\forall e, f(e) \leq c(e)$
- for all vertexes v, $f^-(v) = \sum_{u \in V} f(u \to v)$
- for all vertexes v, $f^+(v) = \sum_{u \in V} f(v \to u)$
- $f^+(v) = f^-(v)$ for all vertices v except for the sink and source nodes s, t

We also define $|f| = f^+(s) = f^-(t)$ as the value of the flow f

6 Tuesday, September 26, 2017

6.1 Flow on a network

So for some reason, Louis also denotes the value of the flow f as ∂f.

There are 2 main questions about network flows, and right now we will discuss the one in the form of the min-cut max-flow theorem.

Starting from any flow, if it is not the maximal flow, we can construct one, but we must define a few terms first

Definition: Cut in a Network

Let G be a network, a cut (S, T) is a partition $V(G) = S \sqcup T$, where $s \in S, t \in T$. The *capacity of the cut* is defined as

$$\sum_{u \in S, v \in T} c(u, v) = c(S, T)$$

Definition: Minimal Cut

Let G be a network, a cut (S, T) is a *minimal cut* if $\forall (S', T')$ cuts of G,

$$c(S, T) \leq c(S', T')$$

Recall: for any flow f and any cut (S, T) on a graph G, $\partial f \leq c(S, T)$.
We can also increase any flow until we get a maximum one from the Ford-Fulkerson algorithm.
And for a maximum flow f' and a minimal cut (S', T'), we get that

$$\partial f' = c(S', T')$$

Which is the min-cut, max-flow equation

The Idea of the Ford Fulkerson Algorithm is that we use the appropriate paths from $s \to t$ to increase the flow of the network to the maximum flow accordingly.

For a network G, a capacity function c and a flow f.

- for any edge (u, v) where $u, v \in V(G)$, the residual capacity of this edge is defined as
$$c_f(u, v) = c(u, v) - f(u, v)$$
- An augmented path is an (undirected) path in the underlying graph G.
- For any edge, the residual capacity of that edge is strictly positive.

Intuition for the Ford Fulkerson Algorithm:

1. if a flow has an augmented path, the flow cannot be maximal if the path goes from s to t.
2. If an augmented path p between s and t exists, then we can adapt p such that ∂f increases and p is no longer augmented.
3. If there exists no augmented path p, this means that the path is maximal
4. We use this construction to argue that there exists an obvious min-cut equal to the resulting value of the flow.

Example, consider any path $p : s \to t$ on graph G, let u, v be any 2 nodes in this path where (u, v) is on the path.

let $m = \{c_f(u, v) : (u, v) \in p\} = \{c(u, v) - f(u, v) : (u, v) \in p\} > 0$ at each edge of the path p, replace $f(u, v)$ by $f(u, v) + m$

Note that ∂f has increased and that p is no longer augmented.

As if $m = c_f(u, v)$, the new flow has the weight of

$$f(u, v) + m = f(u, v) + c(u, v) - f(u, v) = c(u, v)$$

note: Adding m to all edges in p yields an increased flow.

7 Friday, September 29, 2017

7.1 Max-flow Min-Cut Theorem (continued)

The minimal amount needed of edges to remove is the maximum amount of flow of the network.

Consider the capacity function:

$$c : V \times V \mapsto \mathbb{R}$$

$$(u, v) \mapsto \begin{cases} c(u \to v) & \text{if } u \to v \in E \\ 0 & \text{otherwise} \end{cases}$$

Theorem: Conservation

$$\partial f(v) = \sum f^+(v) = \sum f^-(v)$$

$$\partial f(v) = 0, \forall v \neq s, t$$

In particular, the value of f is $|f| = \partial f(s)$.

Min cut Max flow

- For any flow and any cut, $|f| \leq ||(S, T)||$ (this is the flow capacity inequality)

  $|f|$ is max $\longleftrightarrow$ $||(S, T)||$ $\longleftrightarrow$ f is saturated for edges from s to t

- G_f, is the residual graph according to flow f
- If G_f has no path from s to t, then the flow is maximal
- (Ford-Fulkerson Algorithm) If there is a path in G_f from s to t, we can strictly increase the value of $|f|$ on G

Start with the flow capacity inequality: **Lemma** $|f| \leq ||(S, T)||$ for any flow f, cut (S, T)

$$|f| = \partial f(s) = f^+(s) - f^-(s) = \sum_{u \in V} f(s \to u) - \sum_{u \in V} f(u \to s)$$

Observation 1:

$$\begin{aligned}\partial f(v) = 0 : \forall v \neq s, t \longrightarrow |f| &= \sum_{u \in V} f(s \to u) \\ &= \sum_{w} f(w \to o) + \sum_{v \neq s} \partial f(v) \\ &= \sum_{v \in S} (\sum_{u} f(v \to u) - \sum_{w} f(w \to v))\end{aligned}$$

Observation 2:

$$\sum_{v \in S} (\sum_{u} f(v \to u) - \sum_{w} f(w \to v)$$

Assume $u \in S$. For a choice of v, u have the flow $f(v \to u)$ terminate.

$$\begin{aligned}&= \sum_{v \in S} (\sum_{u \in T} f(v \to u) - \sum_{w \int T} f(w \to v)) \\ &\leq \sum_{v \in S, u \in T} f(v \to u) \\ &= ||(S, T)||\end{aligned}$$

Therefore, $|f|$ is maximal iff $||(S, T)||$ is minimal.

Note:

- We say that f saturates an edge if $f(u \to v) = c(u \to v)$
- We say that f avoids an edge if $f(u \to v) = 0$

So then $|f| = ||(S, T)||$ iff f saturates any edge from s to t and avoids any edge from t to s (so all flows from T to S are 0). In this case, the flow is maximal.

So we can say that the flow is maximal if every edge from S to T are saturated and every edge from T to S is avoided.

Definition: Residual Graph

residual capacity:

$$c_f : V \times V \mapsto \mathbb{R} \geq 0$$

$$\begin{cases} c(u,v) - f(u,v) & \text{if } (u,v) \in E \\ f(u,v) & \text{if } (v,u) \in E \\ 0 & \text{otherwise} \end{cases}$$

G_f, the residual graph: Is a directed graph with

$$\begin{cases} V(G_f) = V(G) \\ (u,v) \in E(G_f) \text{ if } c_f(u,v) \neq 0 \end{cases}$$

note: If there is no path from s to t in G_f then Let S be the vertices reachable from s (which also includes s) and let T be the nodes $V(G_f) \setminus S$ (which also includes t). Then (S,T) is a cut on G or G_f

Consider u, v where $u \in S, v \in T$
we know that $c_f(u \to v) = 0$, since if it did, $v \in S$ which would be a contradiction of a cut.

$$c_f(u \to u) = c(u \to v) - f(u \to v) + f(u \to v) = 0 + 0 = 0$$

Conclusion: There are no paths in G_f from s to t, this implies that the flow is maximal. i.e., $|f| = ||(S,T)||$

But what happens if G_f has a path? We used that path $s \to t$ to increase the value of f on G (which is the Ford-Fulkerson Algorithm).

As $s = u_1, u_2, ..., u_n = t$ is an "augmented" path p on G_f where

$$\forall i : c_f(u_i, u_{i+1}) \neq 0$$

Let $F = min(c_f(u_i, u_{i+1}))$ on G, define a new flow f' as follows.

$$f'(u \to v) = \begin{cases} f(u \to v) + F & \text{if } u \to v \in p \\ f(u \to v) - F & \text{if } v \to u \in p \\ f(u \to v) & \text{otherwise} \end{cases}$$

Claim: f' is a flow on G and $|f'| = |f| + F$

1. $\forall(u \to v) : f'(u \leftrightarrow v) \geq 0$ The only problem case where $u \to v$ in the path p is $v \to u$ edge is in G as subtracting F made result in a flow smaller than 0.

   But calculating this out, we know that if $u \to v$ lies in the path p and $F \leq f(u \to v) \forall u \to v$ in the path, so $f(u \to v) - F \geq 0$. So we know there is no problem as the flow always remains equal to or greater than 0.

2. $\forall(u \to v) \in E(G)$ such that $f'(u \to v) \leq c(u \to v)$ but if $u \to v$ in the path, edge in G

$$\begin{aligned} f'(u \to v) &= f(u \to v) + F \\ &= f(u \to v) + c(u \to v) \\ &= c(u \to v) - (c_f(u \to v)) \\ &\leq c(u \to v) \end{aligned}$$

Observation: $|f'| = |f| + F$ since if the orientation changes, there is another adding edge that cancels out the subtracting edge, and otherwise, you add both adding edges.

7.2 (Ford-Fulkerson) Algorithm for finding a maximal flow for a network

1. start with original flow of f=0
2. write down G_f
3. choose a path on G_f from $s \to t$ (if no path exists, the flow is maximal)
4. increase the flow using this path, and repeat step 2 with new flow

8 Tuesday, October 3, 2017

8.1 Applying Min-cut Max-flow

Say we want to prove a statement, that Graph G has a set of edges with property p or size k, as long as any set of vertices with property q has size at least $k4$.

Idea To prove such a statement, we have to use the min-cut, max-flow theorem.

- Turn G into a network, and let the flow value of the graph be k
- The graph should correspond to the set of edges with property p
- The set of k vertices with property Q corresponds to cut of capacity k
- There is a matching in $V(G)$, such that no 2 edges in the set share the same endpoint.

Definition: Vertex Cover

Set of vertices C such that any edge in G has an endpoint of at least one vertex in C.

Theorem: (Konig)

In a bipartite graph, There exists a matching of size k as long as any vertex cover has at least size k.

Note that this variation of the definition of capacity includes ∞. i.e.

$$c : V \times V \mapsto \mathbb{R}^+ \cup \{\infty\}$$

Even in this case, any finite flow has a max flow and a min cut. The point is to go from a bipartite graph to a network as such:

Network	Bipartite Graph
flow	matching
value	size
cut	vertex cover
capacity	size

i.e.

1. Given a bipartite graph with $V(G) = X \sqcup Y$
2. add an s and t node as the source and sink nodes respectively.
3. connect every element of s to x and every element of t to y with an edge.
4. Give a value of 1 to the capacity of the newly added edges, and a value of ∞ to the capacity of the original edges of the bipartite graph.

Conclusion 1: any flow has a flow of $f(u,v) = \begin{cases} 1 \\ 0 \end{cases}$ for any edge $u \to v$

Conclusion 2: if we look at a flow, and the subset of edges from x to y with $f(e) = 1$. This is a matching. Note that any flow yields a matching.

Conclusion 3: Value of that flow is the number of edges in the matching.

Understand cut capacities, Take a cut (S, T), assume finite capacity

$$c(S,T) = \sum_{u \in S, v \in T} c(u \to v)$$

Now, looking at $(X \cap T) \cup (Y \cap S)$, this is a vertex cover for G edges between X, Y such that the endpoints of the edges are not in the vertex cover set C.

The first endpoint lies in $X \cap S$ and the second endpoint lies in $Y \cap T$. Because the edges now goes from S to T contributes to the capacity and $c(e) = \infty$.

so a (S, T) cut implies a vertex cover where the capacity of the cut is equal to the size of the vertex cover.

$$\sum_{u \in T} c(s, u) = \sum_{v \in S} c(v, t) = |C|$$

Conclusion: A cut implies a vertex cover, the capacity of the cut is the size of the cover.

And this finishes the proof by Max-Flow, Min-Cut.

9 Friday, October 6, 2017

9.1 Hall's Theorem on Perfect Matching

Assume $|X| = |Y|$, does there exist a perfect match?

Consider the marriage problem where nodes in X represent women and nodes in Y represent men.

We require that for any k women, they need to be interested (connected by an edge to) at least k men (this is the sufficient condition).

Let G be a bipartite graph, $V(G) = X \sqcup Y$. $|X| = |Y|$.

Marriage Condition

$W \subset X, \Gamma(W) = \{y | (x, y) \in E, \text{ for some } x \in W\}$

Theorem

The following are equivalent:

- G has a perfect matching
- G satisfies the marriage condition.

For a flow f, the value of $|f|$ is the size of the matching, if there exists a flow of value n. If there does not exist a flow of $|f| = n$. Produce a set $W \subset X$ such that $|W| > |\Gamma(W)|$ which implies that the max flow has a value of $k < n$.

There exists a cut (S, T) of capacity $||(S, T)|| = k$

note: $(X \cap T) \cup (Y \cap S)$ is an edge cover of size k

$$|X \cap T| + |Y \cap S| = k$$
$$|X \cap T| + |Y \cap S| = k < |X \cap T| + |X \cap S| = n$$

This is because $|X \cap T| + |X \cap S| = |X|$ as max flow is n.

This implies that $|Y \cap S| < |X \cap S|$.

Take any $W = X \cap S$.

For $W = X \cap S, \Gamma(W) \subset Y \cap S \longrightarrow |\Gamma(W)| < |W|$

Any edge (x, y) with $x \in (X \cap S)$ must have $y \in (X \cap S)$. If however $y \in (Y \cap T)$, edge (x, y) contributes to the capacity of the cut, which in this case is of infinite capacity.
The Theorem is constructive: assume marriage condition is true, a max-flow implies a perfect matching (look at all edges between X anad Y with flow 1).

9.2 Connectivity: Menger's Theorem

Definition: edge-cut

Considering a connected graph G, we define an edge-cut as a set of edges $\{e_1, ..., e_n\}$, such that removing these edges from the graph G disconnects the graph.

Definition: k-connected

We say G is k-connected if the minimal size of the edge cut of G is of size k.

Example: a 1-connected graph means there exists at least one edge that is not lying on a cycle.

Idea: Menger's Theorem uses the Min-Cut Max-Flow theorem to extend this example.

Menger's Theorem

G is k-connected iff any 2 points are connected by k disjoint paths. The max size of a set of paths that are disjoint is k.

This follows from the statement:
If s, t are 2 vertices, what are the minimum amount of edges that need to be removed so that s, t are disconnected from each other? This is the maximal cardinality of the set of $s \to t$ paths.

This will follow from the version of the max-flow min-cut for directed graphs, but now given a directed graph, we create a network with a source node s and a sink node t with a capacity of 1 on all directed edges.

10 Tuesday, October 17, 2017

10.1 Planar Graphs

To define planar and other concepts (i.e. 'faces'), we need a few definitions.

Definition: Open Ball

An open ball around point x with radius ϵ is the set

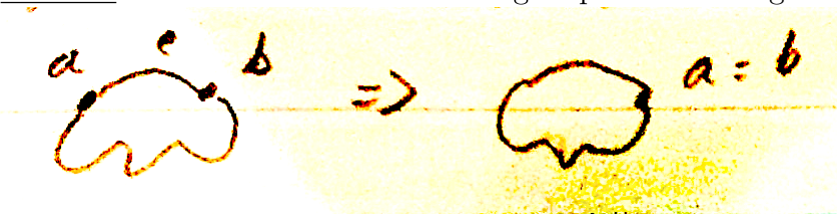
$$B(x, \epsilon) = \{y \in \mathbb{R}^n : ||y - x|| < \epsilon\}$$

Definition: Open set

An open set in $\mathbb{R}^n$ is a set U such that $\forall x \in U$, there exists some ball $B(x, \epsilon) \subset U$ for some $\epsilon > 0$

counter-example: the closed interval $[a, b]$ because for b, there does not exist an $\epsilon > 0$ such that $]b - \epsilon, b + \epsilon[\subset [a, b]$, so $[a, b]$ is not an open set.

More generally, a set that is not open is a closed disk.
Remark: A contraction does not change a planar drawing. For example:



Here we just took edge e and shrank it down. Thus, the planar graph G retains the drawing.

Using these 2 ingredients will help us prove Euler's Theorem.

11 Tuesday, October 24, 2017

11.1 Closed sets

Definition: Closed Sets

For any set $x \in \mathbb{R}^n$, we define the closure of x as follows.

$$\overline{\overline{x}} = \{x \in \mathbb{R}^n : \text{ there exists a sequence in } x, \text{ converging to } x\}$$

In particular: $x \subset \overline{\overline{x}}$, the set is closed if $\overline{\overline{x}} = x$

Example: $[a, b]$ is closed, which converges as

$$a \leq \lim x_i \leq b$$

for any sequence $(x_i) \in [a, b]$

Remark: Closed means any sequence of elements in X that converges has a limit in X

Counter-Example: $]a, b[$, Because we can find a sequence $(x_i)_{i \in l} \in]a, b[$ converging to $b \notin]a, b[$. Take $x_i = b - \frac{1}{i}$

Relation between open and closed sets

The relation between open and closed sets is that

$$X \text{ is open iff } X^c \text{ is closed}$$

Remark: Sets can be open, closed, or neither.
Example: $]a, b[$ is an example of a set that is neither.

12 Friday, October 27, 2017

12.1 More Definitions for Planar Graphs

Curve in $\mathbb{R}^n$

A curve in $\mathbb{R}^n$ is defined as a continuous function

$$\gamma : [a, b] \mapsto \mathbb{R}^n$$

Polygonal Curve

A polygonal curve is defined as a function

$$\gamma : [a, b] \mapsto \mathbb{R}^n$$

where $\exists a = a_0, a_1, ..., a_n = b$ such that $\gamma[a_i, ..., a_{i+1}]$ is a curve.

Drawing of a Graph

For a graph $G(V, E)$, the drawing of a graph is $f : V \mapsto \mathbb{R}^n$ and $f' : E \mapsto$ a polygonal curve in $\mathbb{R}^n$ such that if $e \in E$ is an edge with endpoints a, b, then this implies that $f'(e)$ is a curve from $f(a)$ to $f(b)$.

This, for a planar graph with $n = 2$ and for spherical drawing curves in a sphere, take values in $S^2 = \{x : d(x, 0) = 1\} \subset \mathbb{R}^3$.

Example: this is planar if you look at it directly, also one can see it as a cube with a cube section removed from it.

Note: Drawings of a graph are not invariant. There are multiple ways to draw a cube.

Definition: Crossing of a graph

A crossing of a graph is defined as a point in $f(e) \cap f(e')$, where the point is not an endpoint of either $f(e)$ or $f(e')$

Definition: Region in $\mathbb{R}^n$

An open set $\forall x, y$ in the region, there exists a polygonal curve with endpoints x, y in the region.

Definition: Face

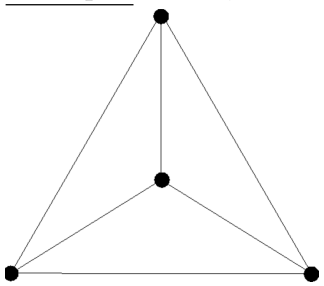
1. Region of a graph drawing
2. Does not interfere with intersect drawing
3. maximal, i.e. another set F' satisfies above properties and $F \subset F$ implies that $F = F'$

Euler's Formula

$$V - E + F = 2$$

For all planar graphs C that has a planar graph drawing, no crossing and must be connected. F refers to the faces of the planar graph drawing.

Example: $V = 4, E = 6, F = 4, 4 - 6 + 4 = 2$



Definition: Jordan Curve Theorem

A curve that divides an area into two distinct regions (one on one side of the curve, and another on the other side of the curve.) Let $[a, b] \in \mathbb{R}^2$ (2 because there is no left or right in space) be a curve. A Jordan Curve is:

1. A closed curve, i.e. $\gamma(a) = \gamma(b)$
2. The curve γ is simple (injective) (no point repeated).

γ divides a region R into 2 maximal regions with the boundary of $\gamma[a, b]$.

Definition: Contraction of an edge

Consider an edge e, the contraction of an edge e of a graph G is the graph $G \setminus \{e\}$ such that for the endpoint nodes a, b of e, $a = b$

12.2 Planar Graphs

12.2.1 Application of Euler's formula for connected, planar graphs

Inequality 1: For a planar graph with $V \geq 3$, $E \leq V - 6$

Proof. Consider an arbitrary planar graph with $V \geq 3$.
For each face, count the number of edges in the face. And since there as at least 3 edges in each face. Plugging this into Euler's formula we get

$$\begin{aligned} 2E &\geq 3F \\ F &\leq \frac{2}{3}E \\ 2 = V - E + F &\leq V - E + \frac{2}{3}E \\ 2 &\leq V - \frac{1}{3}E \\ E &\leq 3V - 6 \end{aligned}$$

■

Example: Consider K_5 (graph with 5 nodes where there is an edge between any 2 vertices).
Note that this graph is not planar as $V = 5, E = 10$ and $10 \leq 3 \cdot 5 - 6$ is not true.

Inequality 2: Assume G has no triangles, then

$$\begin{aligned} 2E &\geq 4F \\ E &\geq 2F \\ 2 = V - E + F &\leq V - E + \frac{E}{2} \\ 4 + E &\leq 2V \\ E &\leq 2V - 4 \end{aligned}$$

Example: $K_{3,3}$, which is the bipartite graph where

$$V = \{x_1, ..., x_n\} \cup \{y_1, ..., y_m\}$$

where any x_i is connected to any y_i by a single edge.

the result is $9 \not\leq 2 \cdot 6 - 4$.

13 Tuesday, October 31, 2017

13.1 Subdivisions

Recall: For a planar connected graph

- $V - E + F = 2$
- $E \leq 3V - 6$ (i.e., K_5 is not planar)
- $E \leq 2V - 4$ (i.e., $K_{3,3}$ is not planar)

Definition: Elementary Subdivision of a graph G

An Elementary subdivision of graph G is a new graph $G \sqcup \{u\}$ where u is placed onto an edge $e = (v, w)$ in the graph, so that the new edges are $(v, u), (u, w)$

Definition: Subdivision of a graph G

A subdivision of graph G is a graph obtained by 0 or more elementary subdivisions of G.

Because of this definition, we can say that:
If a graph is planar, then all of its subdivisions are planar.

A graph containing a subdivision of K_5 or $K_{3,3}$ then implies that the graph is not planar. This implication points both ways as well.

Kuratowski's Theorem

A graph is planar iff it does not contain a subgraph that is a subdivision of K_5 or of $K_{3,3}$

13.2 Dual Graphs

Assume G is planar, now consider a drawing of G in a plane, the vertices of Dual Graph G^* is equal to the faces of G and there is an edge between $v_1, v_2 \in G^*$ if the face v_1 and the face v_2 have a common edge. The dual graph of a graph G depends on the drawing of a graph G, so 2 different isomorphic graphs may have the same dual graph.

We also know that:

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

Counting all the edges coming out of all vertices means that we double count every edge. But what happens when we apply this to G^* instead.

$$G^*(V^*, E^*) \longrightarrow 2E^* = \sum_{v \in V*} d(v)$$

So now this equation means that all the edges that make up every face is being summed up.

13.3 Platonic Solids

A platonic solid is a solid where each face is a regular k-gon and there is the same amount of faces at any vertex.

14 Friday, November 3, 2017

14.1 Classifying Platonic Solids

Platonic Solid: A solid such that each face is a regular k-gon and each vertex is adjacent to l-faces. ex. a tetrahedron or a cube.

All 5 platonic solids:
Tetrahedron – made from triangles – like a Pyramind
Octahedron – Made from two pyramids put onto each other, made from triangles, like a diamond
Hexahedron (Cube)
Dodecahedron – made from pentagons, soccerball
Icosahedron – top and bottom are pyramids made from 5 triangles

Does any platonic solid has a planar graph?
Let us look at the spherical embedding instead, since it does not make sense for the 3D representation to be a planar graph.

Definition: Spherical Embedding

Drawing of the graph $S^2 \subset \mathbb{R}^3$

Geometric Fact: The vertices of a platonic solid lie on a sphere, this means that any platonic solid has a spherical embedding without crossings.

And the graph is spherical without crossing iff the graph is planar.

This can be shown by drawing the stereographic projection, taking a point P at the top of the sphere and a plane tangent to the bottom of the sphere, and the point P will draw a stereographic projection by first intersecting with the edges and vertices of the sphere graph and onto the plane and that is where it is being projected. This produces a planar graph.

This means that the image of a curve is a curve and if 2 curves cross in the plane, this implies that they must also cross on the sphere, and since the graph on the sphere is planar, the projection is also planar.

Recall: For connected and planar graphs

1. $2E = \sum_{F \text{ is a face}} e(F) = kF = lV$
2. V-E+F=2

This implies that

$$V = \frac{2}{l}E, F = \frac{2}{k}E$$

which implies that

$$\frac{2}{l}E + \frac{2}{k}E - E = 2$$

which implies that

$$E(\frac{2}{l} + \frac{2}{k} - 1) = 2$$

which implies that

$$\begin{aligned}
&\longrightarrow \frac{2}{l} + \frac{2}{k} - 1 > 0 \\
&\longrightarrow 2k + 2l - kl > 0 \\
&\longrightarrow -2k - 2l + kl + 4 < 4 \\
&\longrightarrow (k-2)(l-2) < 4
\end{aligned}$$

So this means that $k, l \geq 3$ and that only one platonic solid corresponds to each choice of k, l. So we get the result $3 \leq k, l \leq 5$ which only corresponds to the following pairs: $(3,3), (3,4), (4,3), (3,5), (5,3)$.

14.2 Coloring

Going forward, we will always consider the graph $G = (E, V)$ with gaps.

- A **colouring** by a set C is defined as $c : V \mapsto C$
- This colouring is considered **proper** if no 2 adjecent vertices have the same colour
- Define a **Chromatic Number** $X^{(G)}$, for example:

  $$X^{(Petersen)} = 3$$

  And $X^{(G)} \leq n, n = |V|$ and $X^{(K_n)} = n$

If G has a colouring c, there is an equivalence relation such that

$$v_1 \sim v_2 \text{ iff } c(v_1) = c(v_2) \text{ where } c \text{ is a colouring}$$

So we have the equivalence class $\overline{\overline{V}}$ is the set of all vertices with the same colour.

And if the colouring is proper, then the equivalence classes are independent as subgraphs.

But what is the minimal size of a set c with a proper colouring?

conversely: say V is partitioned into subsets $C_1, C_2, ..., C_k$ where

$$V = \sqcup_{i=1}^{k} C_i$$

Where each C_i is the independent subgraph such that

$$c : C \mapsto \{1, 2, ..., k\} \text{ where } v \mapsto i \text{ if } v \in C_i$$

14.3 Bounds for colouring

Recall that a clique is a set of vertices whose induced subgraph is complete.

14.3.1 Lower Bound

Define $W(G)$ as the **size of the largest clique in** G.

> note: $X(G) \geq W(G)$ because of G is complete, then that implies that $X(G) = W(G)$.

Argument: Assume G has a clique of size $l : v_1, ..., v_l$ then the subgraph induced by $v_1, ..., v_l$ has the chromatic number l where $l = W(G)$
Now note that $l = W(G)$.
Since $X(G) \geq X(G')$ for any subgraph G'.
Define $\alpha(G)$ as the size of the longest independent set of vertices.

$$X(G) \leq \frac{n}{\alpha(G)}, n = |V|$$

say $X(G) = l \longrightarrow V$ can be paritioned into l different independent sets, i.e. l different sets of size $\leq \alpha(G)$
$\longrightarrow n \leq l, \alpha(G) = X(G)$.

14.3.2 Upper Bound

Start with greedy colouring (following is the steps for a greedy colouring algorithm).
$V = \{v_1, ..., v_n\}$

- Assign colour 1 to v_1
- If v_2 is not adjacent with v_1, colour it 1, otherwise, colour it 2
- If v_3 is not adjacent to v_1, then colour it 1, otherwise if not adjacent to v_2, colour it 2, otherwise, colour it 3.
- This goes on for every $v_i, \forall i : 1 \leq i \leq n$

Definition: Largest degree of a vertex in the graph G

$\Delta(G)$ is the largest degree of a vertex in the graph G, note that $X(G) \leq \Delta(G) + 1$

Brook's Theorem

$X(G) \leq \Delta(G)$, unless G is an odd cycle or K_n

15 Tuesday, November 7, 2017

15.1 Colouring Problems

Consider the function $c : V(G) \mapsto C$

15.1.1 Lower Bound

Recall that $\Delta(G)$ is the largest degree of a vertex in the graph G.
It is obvious to use that $\Delta(G) \leq X(G)$
So what is the role of $\Delta(G)$?

We know for the upper bound that the greedy algorithm gives us a colouring with the smallest amount of colours. From that algorithm we concluded that G has a colouring of $u + 1$ colours as long as no u predecessors are adjacent to v_i for any i.

In particular, G always has a colouring of $\Delta(G) + 1$.

Brooke's Theorem

$$X(G) \leq \Delta(G) + 1 \text{ and } X(G) \leq \Delta(G) \text{ iff } G \neq \begin{cases} K_n \\ \text{an odd cycle} \end{cases}$$

We are only going to look at 2 cases when $\Delta(G) = k$:

1. Not all vertices have degree k
2. the graph has a cut-vertex

(For 1), the idea is that we will used the greedy colouring algorithm with a clever ordering of the nodes. We will order the vertices $v_1, ..., v_n = V$ such that $d(v) < k$, so there are at most $k - 1$ predecessors who are adjacent for each vertex.

Definition:Trees

Trees are a connected acyclic graph where any 2 vertices are connected by a single path.

Definition: Spanning Tree

A spanning tree is a choice of edges such that the underlying graph is a tree for all nodes.

Note: Any connected graph G has a spanning tree.

Proof. (Proof any connected graph G has a spanning tree)
Use induction on the number of edges, which we will denote by l.

1. $l = 1$
2. Assume any connected graph of $|E| \leq l$ has a spanning tree, Now consider the graph with $l + 1$ edges. We can pick any $e \in E(G)$. Now consider the case when e does not lie on a cycle, that would mean that $G \setminus e$ is not connected, which implies that $G \setminus e$ has 2 connected conponents $G_1 \sqcup G_2$, which implies that each graph has their own spanning trees (T_1, T_2 respectively) since they both have $\leq l$ edges. And to finish this proof, we need to show that $T_1 \cup T_2 \cup e$ is a spanning tree.

■

16 Tuesday, November 14, 2017

16.1 5-Colour Theorem

Recall: that the **6-colour theorem** stating that every planar graph has a 6-colouring is easy because by Euler's formula, there exists a vertex with degree ≤ 5, so this implies the result.

So what about the 5-colour theorem?

Consider the graph with the node u connected to 5 nodes, labelled a, b, c, d, e clockwise. Assume that all nodes connected to u all have a distinct colour (Lets say $c(a) = 1, c(b) = 2, c(c) = 3, c(d) = 4, c(e) = 5$). Now consider the subgraph of all vertices, coloured $1, 3$ connected to the node c (we will call this subgraph G_1), and consider the subgraph of all vertices, coloured $1, 3$ connected to the node a (we will call this subgraph G_2).

Scenario 1: vertices a, c are not connected, because if so, then G_1, G_2 are not disjoint in G, but we can make it work by simply alternating the colouring of the nodes of one of the trees from $3 \leftrightarrow 1$.

If we remain with a proper colouring by colouring the node u with the colour 3. This is only not proper if there exists a node v connected to G_1 that is the same colour as the node it is connected to. This is impossible as v would be in $V(G_1)$ and would not be the same colour as it would have its colour changed.

Scenario 2: vertices a, c are connected, so we cannot change the colours. Now we will look at:

- G_2 : subgraph of $\forall u \in V(G_2)$ connected to node b
- G_4 : subgraph of $\forall u \in V(G_4)$ connected to node d

but now these graphs must cross an edge but cannot connect, so *scenario 1* must apply to G_2, G_4.

Question 1: What is the smallest proper colouring we can find?
Question 2: How many colourings are there?

Consider the birthday problem. On a graph, its assigning colours $\{1, ..., 365\}$ to nodes to represent n people. (K_n).

For n people, it's colouring of K_n is the possible combinations, where there is an edge between 2 nodes if they have a unique bday.

For a room of 23 people, the probability of 2 people sharing the same birthday is 52%, and 70 people results in a probability of 99%.

This is why we may be interested in how many colourings there are.

Definition: Chromatic Polynomial

For a graph G and an positive integer t,

$$P(G,t) = \text{ The number of proper } t\text{-colourings on } G$$

Example 1: For n vertices without any edges,

$$P(G,t) = t^n$$

Example 2: For K_n,

$$P(K_n,t) = \frac{t!}{(t-n)!}$$

Why is $P(G,t)$ a polynomial and how to compute it?

Recall that contracting an edge $e = (u,v)$ is removal of the edge e and $u = v$. Where the graph with the edge e is contracted is graph G_e.

$$P(G,t) = P(G \setminus e, t) - P(G_e, t)$$

The differences between $G \setminus e$ and G_e is that u, v can be the same colour in $G \setminus e$ but not in G. But in G_e, u, v are the same node and therefore the same colour, so the equation holds.

$P(G,T)$ is a polynomial

Proof. (Induction on the number of edges)
Base Case: If there are no edges, we know that $P(G,T) = t^n$

Induction Step: G_e and $G \setminus e$ are polynomials, and a polynomial subtract a polynomial is still a polynomial. ■

17 Friday, November 17, 2017

17.1 Graph Colouring

Recall: Chromatic Polynimals are $P(G,t)$.

There are 2 applications:

1. Supposed we want the Chromatic number of the cycle $P(C_3,t) = t(t-1)(t-2)$
2. $C_4 : P(C_4,t) = t(t-1)^2(t-2) + t(t-1)^2$
   For C_4, there are 2 ways to colour it. Treat pairs in a bipartite split with different colours (2 colours) and setting all of the vertices as different colours.

General Result for Cycles:

$$P(C_n,t) = (t-1)^n + (-1)^n(t-1)$$

Proof that result agrees with above:

$$e \in C_n, C_n \setminus \{e\} = \Pi_n \to (C_n)_e = C_{n-1}$$

Proof. (Proof by Induction)
Base Case: $n = 3$, $P(C_3,t) = t(t-1)(t-2) = (t-1)^3 - (t-1)$ **Induction Step:** By induction, assume the format is true for any n-cycle.
Then:

$$\begin{aligned} P(C_{n+1},t) &= P(\Pi_{n+1},t) - P(C_n,t) \\ &= t(t-1)^n - ((t-1)^n + (-1)^n(t-1)) \\ &= (t-1)^n(t-1) - (-1)^{n+1}(t-1) \\ &= (t-1)^{n+1} - (-1)^{n+1}(t-1) \end{aligned}$$

■

This also applies for any tree (which are all connected and acyclical).

> Remark: Any tree has a vertex with degree > 1 (if $|E| > 1$)

Proof. (Proof by induction on the number of edges)
Base Case: For a connected tree with only 2 edges, one vertex must have a degree of 2.
Induction Step: Assume this holds for all trees with $|E| = n$, Consider a tree with $|E| = n + 1$. Just remove any edge e, so $G \setminus e$ is a union of trees with less than n edges, and therefore must have the statement be true.
Therefore by induction, every tree has a vertex of at least degree 2. ■

Let e be an edge with endpoint node v ($deg(v) = 1$). Then

$$T_n \setminus \{e\} = T_{n-1} \sqcup \{v\}, (T_n)' = T_n''$$

Proof. (Proof by induction on edges)
Base Case: $E = 1$, $P(T_2, t) = t(t-1)^{2-1}$
Induction Step: Assume formula holds for any tree with n vertices.
Then

$$\begin{aligned} P(T_{n+1}, t) &= P(T_n \sqcup \{v\}, t) - P(T_n'', t) \\ &= tP(T_n', t) - P(T_n'', t) \\ &= t(t(t-1)^{n-1}) - t(t-1)^{n-1} \\ &= t(t-1)^n \end{aligned}$$

■

17.2 Spanning Trees

motivating question: How should Toronto plow snow as efficiently as possible?

i.e. For a graph G, highlight edges such that:

- Each vertex in the graph is some endpoint for some edge
- There does not exist a path between 2 vertices

Definition: Spanning Tree

For a connected graph G, a spanning tree is a subgraph T which is a tree, $V(T) = V(G)$

Definition: Weighted graph

A weighted graph is a graph G with a weight function

$$w : E(G) \mapsto \mathbb{N}$$

Definition: Minimum Spanning Tree

A minimum spanning tree is a spanning tree where the sum of weights on edges is minimal among all possible spanning trees.

17.2.1 Finding a Minimum Spanning Tree

1. Prim's Algorithm
2. Kruskal's Algorithm

Kruskal's Algorithm

Kruskal's Algorithm is a greedy algorithm, meaning that it takes the most optimal decision at each step individually to get the most desired result once the algorithm concludes.

1. Order edges by increasing edge weight
2. Highlight lowest value edges such that no cycle is created
3. Will end up with a minimum spanning tree once all nodes are covered

Assume $T \subset G$ is a spanning tree.
$e \in E(G \setminus T) \longrightarrow T \cup \{e\}$ is no longer a tree (we have introduced a cycle), so $T \cup \{e\}$ has a unique cycle (since T initially did not have a cycle).

Proof. (Proof of result)
We first observe that:

1. The algorithm results in a spanning tree

2. Any choice of edges in the algorithm is part of a minimum spanning tree

So lets look at these observations. We know the result is acyclic because the algorithm does not choose edges that create cycles. We also know that the result is connected because if no, pick 2 connected components in G, the components are connected at some point an edge with minimum weight will connect components in T. ■

18 Tuesday, November 21, 2017

This lecture was simply looking at different algorithms for looking for minimum spanning trees and an example of them running on a graph.

18.1 Kruskal's Algorithm

This algorithm is essentially ordering all the edges by edge weight, and then adding the edge one by one if they did not create a cycle. We continue adding edges until we get a Minimum Spanning Tree

18.2 Prim's Algorithm

We start by picking a node and we choose the smallest connected edge until we get a Minimum Spanning Tree

18.3 Dijkstra's Algorithm

We find the minimum weight of a $u \to v$ path for a graph G with nodes $u, v \in V(G)$.

19 Friday, November 24, 2017

19.1 Spanning Cycles – Eulerian Cycles

Let a graph G have a cycle C where $E(G) = C$ and $V(G) = C$.

Definition: Eulerian Path /Cycle

An eulerian path is a path that goes through each edge once. An eulerian cycle is an eulerian path that is a cycle.

Examples of eulerian cycles:

- Joint cycles
- A single cycle

Supposed G is a connected graph, if an eulerian C is plausible on G, then G can be decomposed into cycles.

Definition: G can be decomposed into cycles

if $E(G) = E(C_1) \sqcup ... \sqcup E(C_n)$ where each C_i is a simple cycle.

(For now) Assume:

Theorem G is Eulerian iff G can be decomposed into cycles.

Theorem G is Eulerian iff all vertices have an even degree.
Proof Checklist:

1. G Eulerian implies all vertices have an even degree
2. All vertices have an even degree implies G is eulerian
3. G Eulerian iff G can be decomposed into cycles.

Note: An even graph is a graph with nodes all of an even degree.

Lemma If the degree of all vertices is ≥ 2 then v contains a cycle.

Proof. Take a maximal path between 2 vertices u, v so that $p : u = v_1, ..., v_n = v$ and maximal in the sense that there is no edge between v_n and some v_{n+1}, such that $v_1, ..., v_n, v_{n+1}$ is a path. ■

Going down proof checklist:

1. G is Eulerian implies that evert vertex degree is even goes back to the 7 bridges problem.

2. if all vertices have an even degree, this implies that G is eulerian.

   Proof. Induction on n simple cycles in G where $n \geq 1$.
   Base Case: $n = 1$, is trivial
   Induction Step: Assume this works for k simple cycles.
   G is eulerian implies that G is an even graph containing k simple cycles, take a graph G without even degree vertices, which contain $k+1$ simple cycles. If we remove simple cycle C, then $G \setminus C$ has k simple cycles and even degree vertices. ■

20 Tuesday, November 28, 2017

20.1 Hamiltonian Paths /Cycles

Definition: Hamiltonian Path /Cycle

An hamiltonian path is a path that goes through each node once. An hamiltonian cycle is an hamiltonian path that is a cycle.

Note: A Hamiltonian cycle implies a hamiltonian path since a hamiltonian cycle is a path.

Example: A cycle is an example of a Hamiltonian Cycle

- G is hamiltonian implies G' is hamiltonian if $V(G') = V(G)$ and $E(G') \supseteq E(G)$
- A non example would be a cycle that can be decomposed into 2 simple cycles.
- Another example would be K_n

If G is bipartite, G may be Hamiltonian.
If G is bipartite and hamiltonian, that implies that $|X| = |Y|$ where $V(G) = X \sqcup Y$

20.1.1 Necessary Conditions

Assume G has a hamiltonian cycle. This means that G is at least 2-connected.

20.1.2 Sufficient Conditions

Theorem if $d(v) \geq \frac{|V|}{2}$ for all vertices $v \in V$ then this implies that G has a Hamiltonian Cycle (for $|V| \geq 3$).

note: This bound is tight, there exists graphs where $d(v) = \frac{|V|}{2} - 1$ that are not Hamiltonian.

Consider a hamiltonian graph G with adjacent vertices u, v implies that $G + uv$ is Hamiltonian. The converse is only sometimes true.

21 Friday, December 1, 2017

21.1 Hamiltonian Graphs

Definition: Hamiltonian Graphs

A graph $G = (V, E)$ contains a a Hamiltonian cycle (cycle with the vertex set V).

This is different when compared to an Eulerian graph.

Let's say G is Hamiltonian, add an edge e anywhere, $G \cup \{e\}$ is Hamiltonian.

Let's say G is Hamiltonian, add an edge e between non-adjacent nodes, $G \cup \{e\}$ is never Eulerian.

Consider G as an even graph ($d(v)$ is even for all $v \in V$)

21.1.1 Necessary Condition

The graph G is bipartite and $|X| = |Y|$ as well as 2-connected.

21.1.2 Sufficient Condition

Dirac's Theorem That $\forall v \in V, d(v) \geq \frac{|V|}{2}$ implies that the graph G is hamiltonian

remark: For 2 adjacent vertices u, v, $d(v) + d(u) \geq |V|$, this implies that $G \setminus \{e = (u, v)\}$ remains Hamiltonian

Ore's Theorem For 2 adjacent vertices u, v, $d(v)+d(u) \geq |V|$, this implies that $G \setminus \{e = (u, v)\}$ remains Hamiltonian (rephrase of Dirac)
Now consider the graph G, $\forall u, v \in V$ that are non-adjacent, if $d(u) + d(v) \geq |V|$, Consider $G \cup \{e = (u, v)\}$. Repeat until no such edges remain, this is the **closure** of G

Bondy-Chvatal Theorem

G is hamiltonian iff $\overline{\overline{G}}$ is hamiltonian