Lecture Notes for Graph Theory

Emulie Chhor

May 30, 2021

Introduction

1. Fundamentals Concepts: Types of Graphs, Path/Cycle, Degrees
2. Trees
3. Connectivity
4. Optimization
5. Shortest Path: Trails, Circuit, Path and Cycles
6. Planar Graphs
7. Flow
8. Coloring
9. Matching
10. Ramsey Theory

1 Fundamentals Concepts: Graphs, Digraphs, Degrees

1.1 Why study Graph Theory

TODO

1.2 Overview

1. What is a graph
2. Terminology: walk, trail, path, circuit, cycle
3. Graph Cycle
4. Connected Vertices and Connected Graphs

5. Types of Graphs: Path Graph, Cycle Graph, Complete Graph, Complement of a graph, Bipartite Graph, Complete Bipartite Graph
6. Directed Graphs
7. Degree of a Graph

1.3 What is a Graph

Un graphe est un "ordered pair" composé de deux éléments:

1. Vertex: ensemble des "noeuds" composants le graphe
2. Edges: ensemble de sous-ensembles qui nous dit quels "noeuds" sont reliés

Notre but est de différentier les différents types de graphes et de définir la terminlogie pour parler d'un graphe

1. Undirected Graph vs Directed Graph
2. Simple Graph
3. Order, Size
4. Adjacence

Definition 1.3.1 (Graph). *A graph G is an ordered pair G=(V,E) where V is a finite set of elements and E is a set of 2 subsets of V*

Definition 1.3.2 (Undirected Graph). *An undirected graph is a graph whose edge subsets are not ordered. In other word, if two nodes are connected, then we can reach a to b and b from a.*

Definition 1.3.3 (Directed Graph). *A directed graph, also called digraph, is a graph that has a direction associated with its edges. In other words, the subsets in the Edge set are ordered. The edges are called arcs.*

1. *Out Degrees: Number of vertices comming out from x noted* $od_G(x)$
2. *In Degrees: Number of vertices comming in to x noted* $id_G(x)$

Definition 1.3.4 (Multigraph and Pseudographs). *A multigraph is a graph G=(V,E) is an undirected graph where the edges set is a multiset, which means that there can be multiple edges between two vertices. The number of distinct edge is called the multiplicity*

Definition 1.3.5 (Order and Size).

1. *Order —V— : number of vertex in the graph*
2. *Size —E—: number of edges in the graph*

Definition 1.3.6 (Simple Graph).

1. *No loop*
2. *No multiples edges*

Definition 1.3.7 (Adjacence)**.** *On peut parler d'adjacence pour les vertex et les edges.*

1. *Vertex Adjacence: 2 vertex are adjacents if they are connected by an edge*
2. *Edge Adjacence: 2 edges are adjacent if they have a vertex in between them*

Theorem 1.3.1. *The sum of the degree of all vertices is an even number*

$$\sum deg(v)$$

Plus généralement, the sum of the degrees of all vertices is twice the number of edges

$$\sum deg(v) = 2|E|$$

1.4 Terminology

Definition 1.4.1 (Walk)**.**

1. *Walk: Sequence of adjacent vertices. We can go back on our steps: we can traverse edges and vertices several times. We say the vertices lie on the walk.*
2. *Length: Number of "steps" we make (even though we may go back and forth).*
3. *Open walk: the final vertex is not the same as where we started*
4. *Closed Walk: the end vertex is the same where we started*

Remarque. *On peut utiliser les définitions suivantes pour les trail et autres aussi:*

1. *open/closed*
2. *endpoints*
3. *length*

Definition 1.4.2 (Trail)**.** *A sequence of adjacent vertices without traversing the same edge more than once*

Definition 1.4.3 (Path)**.** *A path is a sequence of adjactent vertices, but we cannot traverse the same vertices more than once (which also means we can't traverse the same edge). Can be defined as*

1. *List of vertices:* $P = (v_1, v_2, ..., v_8)$
2. *List of alternating vertices and edges:* $P = (v_1, v_1v_2, ..., v_8)$

Habituellement, on préfère définir un chemin par une liste de vertices

Definition 1.4.4 (Circuit)**.** *Closed trail of length 3 or more*

Definition 1.4.5 (Cycle)**.** *Closed path that has a length greater than or equal to 3. Sometimes, the definition differ and we may traverse vertex several times (but can't cross edges).*

Definition 1.4.6 (Path and Cycle). *1. A Path P_n is a graph whose vertices can be arranged in a sequence such that the edge set is $E = v_i v_{i+1} | i = 1, 2, ..., n-1$*

2. A Cycle C_n is a graph whose vertices can be arranged in a cyclic sequence such that the edge set is $E = v_i v_{i+1} | i = 1, 2, ..., n-1 \cup v_1 v_n$

Definition 1.4.7 (Degree of Path and Cycle). *The degree of a path and a cycle is the number of vertex it has.*

Definition 1.4.8 (Girth). *Smallest Cycle in the graph*

Definition 1.4.9 (Distance and Diameter between vertices). *Soit deux noeud u et v.*

1. *Distance entre u et v: plus court chemin entre u et v*
2. *Diameter entre u et v: plus long chemin entre u et v*

Theorem 1.4.1 (Properties of Degrees in Path and Cycle). *1. A path of degree n has n nodes and (n-1) edges*

2. A cycle of degree n has n nodes and n edges

Proposition 1.1. *Every graph G contains a path of length n and a cycle of length at least n+1*

1.5 Connected and Disconnected Graphs

Definition 1.5.1 (Connected Graph). *A graph is connected if for every pair of disinct vertices $u, v \in V(G)$, there is a path from u to v in G. Otherwise, we say the graph is disconnected*

Definition 1.5.2 (Connected Vertices).

Definition 1.5.3 (Open and Closed Neighborhood). *TODO*

1.6 Families of Graph and Special Graph

1. Complete Graph K_n: simple graph with an edge between every pair of vertices
2. Empty graph: Graph with no edges
3. Bipartite Graph: a graph whose vertex can be partitionned into two sets V_1 and V_2 such that every edges $u, v \in E$ has $u \in V_1$ and $vinV_2$
4. Complete Bipartite Graph: every node can reach all nodes in the other subset (end)
5. Star
6. k-regular graph: each vertex is degree k
7. Cubic Graph: 3-regular graph (ex: Petersen Graph)

8. Irregular graph: all of its vertices have distinct degrees. There exist only one irregular graph, the graph made up of a single vertex
9. Path Graph:
10. Cycle Graph:
11. Hypercube Graph

1.6.1 Bipartite Graphs

Definition 1.6.1 (Bipartite Graph)**.** *A graph is bipartite if we can split that graph in two sets such that all vertices in A maps to B*

Definition 1.6.2 (Complete Bipartite Graph)**.** *A complete bipartite graph is a bipartite graph where all vertices in A maps to all vertices in B*

Theorem 1.6.1 (Bipartite graph and odd cycle)**.** *A graph is bipartite $\iff$ it has no odd cycle*

If a graph is bipartite, then it has no odd cycle. The proof is done by contradiction

1. Let G be a bipartite graph and c be an odd cycle such that $c = (v_1, v_2, ..., v_n, v_1)$
2. Because G is bipartite, then we can partition odd vertices into set X and even vertices into set Y (because adjacent vertices cannot be in the same component)
3. Since c is an odd cycle, then v_n is odd.
4. This contradicts the fact that G is bipartite, because two adjacent vertices are in the same component ($v_1 and v_n$) are adjacents and both odd.

□

If a graph has no odd cycle, then it is bipartite. The proof is also done by contradiction

1. Let G be a graph with no odd cycle.
2. Let's partition the vertices of the cycles by its parity such that

$$X = v \in V(G)|d(v, w) \text{ is even}$$

$$Y = v \in V(G)|d(v, w) \text{ is odd}$$

$$X \cap Y = \text{ (distance is unique)}$$

$$X \cup Y = V(G) \text{ (connected graph)}$$

3. SFC, there are two adjacents vertices that are in the same set: $a, b \in X$ or $a, b \in Y$ such that $ab \in E(V)$

4. Suppose a=w, then d(a,w)=0 (the distance is even). Thus, d(b,w) is even and d(a,b) is even. However, since a,b are adjacents then d(a,b)=1. Therefore, $a \neq b \neq w$
5. Consider the shortest path from aw denoted by P, the shortest path from bw denoted by Q, and m be the last common vertex of P and Q. Let P_1 and Q_1 be the path from a to m and b to m respectively and P_2 and Q_2 both be the path from m to w
6. Then $|Q_1| = |P_1|, |P| = |P_1| + |P_2|, |Q| = |Q_1| + |Q_2|$. Since a,b are in the same set, then d(a,w) and d(b,w) must have the same parity. Since $|P_1| = |Q_1|$ by construction, then $|P_2|$ and $|Q_2$ must have the same partity by parity of integers
7. If we construct a cycle from M to A to B using d(a,b), P_2. Q_2, we have an odd cycle: $|P_2| + |Q_2| + 1 = 2k + 1 \forall k \in \mathbb{Z}$, which contradicts the fact that G has no odd cycle

□

1.6.2 Complete Graph

Theorem 1.6.2. *Let G=(V,E) be a graph with m vertex. Alors, la somme de tous les degrés d'un graphe est le double du nombre de edges.*

$$\sum deg(v) = 2|E| = 2m$$

Remarque. *La preuve se fait par induction. On suppose qu'on a un graphe de m+1 edges et qu'on lui enlève un dege arbitraire.*

Theorem 1.6.3 (Handshaking Theorem). *The Number of Edges in a Complete Graph is* $|E| = \frac{N(N-1)}{2}$ *Proof:*

1. *N vertex in graph*
2. *The degree of each vertex is N-1 by definition of a complete graph (each node is connected to all the other)*
3. *Sum of all vertex degrees is* $\sum d(v_i) = N(N-1)$
4. *Number of edges is* $|E| = \frac{N(N-1)}{2}$ *because we counted all edges twice*

1.6.3 Irregular Graphs

Definition 1.6.3 (Irregular Graph). *A graph is irregular if all of its vertices have different degrees*

Proposition 1.2 (The single vertex graph is the only irregular graph). *The proof is done by contradiction by showing that there cannot be a vertex adjacent to n-1 vertices and one adjacent to 0 vertex.*

1. *Let G be a graph with n vertex with different degrees, where the degrees of each vertex is between* $0 \leq deg(v) \leq n-1$
2. *Because each vertice has a different degree, this must means that there exist a vertice adjacent to 0 vertex and one adjacent to n-1 vertices.*
3. *However, the last statement is a contradiction, because if a vertex is adjacent to none, then there are only n-2 vertices left.*

1.6.4 Complement of a Graph

Definition 1.6.4 (Complement of a Graph)**.** *Let G be a graph. The complement of G, noted* $\bar{G}$*. uv is an edge of* $\bar{G} \Longleftrightarrow$ *uv is not an edge of G*

Definition 1.6.5 (Self Complementary Graph)**.** *A graph that is isometric to its complement is self complementary*

Theorem 1.6.4 (Connectivity of the complement of a graph)**.** *A graph or its complement must be connected* $\Longleftrightarrow$ *if a graph is disconnected, then its complement must be connected*

Connectivity of the complement of a graph. Let's work with the second proposition

($\Longrightarrow$) If G is a disconnected graph and u,v be two vertices in G such that u and v are in different components. Then, by definition, uv must be in $\bar{G}$. Therefore, uv $\in \bar{G}$, a connected graph.

2. ($\Longleftarrow$) If uv $\in V(G)$, such that they are in the same components. Then u,v are adjactents. If uv are not in the same component, there exist w such that $uw \in V(G)$ and $vw \in V(G)$, therefore, there exist a uv path in $\bar{G}$

□

Corollary 1.6.1 (There is not disconnected self complement graph)**.** *If there were such a graph, then G and* $\bar{G}$ *must be disconnected, which contradicts the fact that G or* $\bar{G}$ *must be connected.*

1.7 Degrees of a Graph

Definition 1.7.1.

1. *minimum degree:* $\delta(G)$
2. *maximum degree:* $\Delta(G)$
3. *Isolated Vertex: deg(G)=0*
4. *End Vertex (leaf): deg(G)=1*

Theorem 1.7.1 (Every Graph has an even number of odd Degree vertices)**.** *Proof by contradiction by using the fact that sum of odd number is even and Handshaking*

1. *Let G be a graph with odd number of odd degree vertices. Let X and Y be the sets of even and odd vertices respectively such that $X = v \in G(V)|$ deg(v) is even , $Y = v \in G(V)|$ deg(v) is odd*

2. *Remark that*
   - $\sum_{v \in G(V)} = 2m$ *because of Handshaking Theorem*
   - $\sum_{v \in X} = 2k$ *(even) because the sum of even number is even*
   - $\sum_{v \in X} = 2k$ *(even) because the sum of odd number is even (where our contradiction lies)*

3. *However,* $\sum_{v \in Y} = \sum_{v \in G(V)} - \sum_{v \in X} = 2(k-l)$ *is even, which contradicts the facts that* $\sum_{v \in Y}$ *is odd.*

Theorem 1.7.2 (Degree sum condition for connected graph)**.** *Let G be a graph of order n. If* $deg(u) + deg(v) \geq n - 1$*, then G is connected and* $diam(G) \leq 2$

Proof. Pour montrer que G est connexe, on veut montrer qu'il existe un chemin entre d'un vertex u à v. Si u et v sont adjacent, alors c'est trivial. Si u et v ne sont pas adjactents, on utilise l'hypothèse $deg(u) + deg(v) \geq n - 1 \geq n - 2$ nous dit qu'il existe n-1 edges qu'il existe un vertex intermédiaire w par lequel on peut passer pour se rendre à v, et donc qu'il existe un chemin (u,w,v) de deg(2). □

Theorem 1.7.3 (Minimum Degree Condition for Connected Graph)**.** *If G is a graph of order n with* $\delta(G) \geq \frac{n-1}{2}$*, then G is connected. Note:* $\delta(G)$ *is the minimum degree of a graph*

Proof. On veut utiliser le degree sum condition for connected graph.

1. $deg(u) + deg(v) \geq \delta(G) + \delta(G)$
2. $deg(u) + deg(v) \geq \frac{n+1}{2} + \frac{n+1}{2}$ par hypothèse
3. $deg(u) + deg(v) \geq n - 1$, which is always true by degree sum condition

□

Remarque (Necessary vs Sufficient condition)**.** *The minimum degree condition for connected graph is sufficient to say if a graph is connected, but not necessary.*

1.8 Isomorphic Graph

Definition 1.8.1 (Isomorphic Graph)**.** *Two graphs are isomorphic if they have the same structures ie we can match each vertices in graph G to each vertices in graph H. We want to "rename" the vertices.*

$$\phi : V(G) \to V(H)$$

Formally, we say that two graphs G and H are isomorphic if there exists a bijection

$$\phi : V(G) \to V(H)$$

such that $uv \in E(G) \iff \phi(u)\phi(v) \in E(H)$ *We write* $G \cong H$

Remarque. *Isomorphic function is bijective which means that it is*

1. *injective: for each image, we can find a unique x* $f(a) = f(b) \implies a = b$
2. *surjective: we can reach all images from the domain*

We may prove that two graphs G and H are not isomorphic if they don't have the same number of vertices and/or edges.

Definition 1.8.2 (Degree of Sequence)**.** *The degree of a sequence is the ordered set of degree(v)*

1. *non-increasing:* $d_n \geq d_{n+1}$

Theorem 1.8.1 (Isomorphic Graphs have the same degree sequence)**.** *If two graph are isomorphic, then they have the same degree sequence. The converse is not true.*

Proof. TODO □

Remarque. *To make the degree sequence unique, we may require that the degree sequence be non increasing.*

Problème.

1. *Determine if a degree sequence is a graph:*
   - *Number of odd degrees vertices must be even*
   - *Use Handshaking Lemma: sum of vertex degrees is twice the number of edges + parity of sum of degrees*
   - *Use degree sum of connected graph: a single vertex cannot be adjacent to more than n-1 vertex*
2. *Draw a graph with the degree sequence*

Remarque (Algorithm to find if degree sequence is graph)**.** *We can compare the original degree sequence to another one we can construct using the following algorithm:*

1. *remove largex degree vertex* v_0
2. *substract 1 to the next* $deg(v_0)$ *vertex in the non-incresing degree sequence.*

2 Connectivity

2.1 Overview

1. Fundamentals of Connectivity
2. Bridges
3. Minimum Spanning Trees
4. Vertex Cuts and Connectivity
5. Edges Cuts and Connectivity
6. Menger Theorem
7. Eulerian and Hamiltonians path/cycles

2.2 Fundementals of Connectivity

Definition 2.2.1 (Connected Vertices). *Two vertices are connected if there exist a path between them*

Definition 2.2.2 (Connected Graph). *A graph is connected if, for every vertices in the graph, we can reach any other node. If a graph is not connected, we say it is disconnected made of components.*

Remarque. *A connected graph has one single component*

Definition 2.2.3 (Components of a graph). *A component of a graph is a maximal connected subgraph, which means that*

1. *Connected: All the nodes in the subgraph can be reached from one to another*
2. *Maximal: there no is node or vertex that we can add without violating the connected property*

The number of component in a graph is noted K(G)

Theorem 2.2.1 (Connected Graph contains Two Non-cut Vertices). *WHAT*

2.3 Bridges

Definition 2.3.1 (Edge Substraction). *Let G be a graph and db be an edge in the graph G. The graph G-e is the graph G in which we remove the edge db. If we remove several edges from G, we can write* $G - e_1, ..., e_n$

Definition 2.3.2 (Bridge). *An edge* $e \in E(G)$ *is a bridge if removing that edge from the graph creates a new component In other word, removing the edge makes G-e disconnected. Formally, we write* $K(G) = K(G - e) - 1$

Definition 2.3.3 (Cut Vertex). *Let G be a graph and v be a vertex of that graph. If removing v from G disconnect its graph, then we say it is a cut vertex.*

Theorem 2.3.1 (An edge is a bridge iif it lies on no cycle)**.** *Let G be a graph and e be an edge in G. The edge e is a bridge* $\iff$ *it lies on no cycle.*

Proof. To prove the previous statement, consider the contrapositive: if the edge lies on a cycle, then it is not a brigde.

1. Consider a cycle C on the graph G such that $C = (c_0 = u, ..., c_i = v, ...c_n = u)$
2. Because C is a cycle, then there exist two u-v path, let's say e and P.
3. Spdg, let's say we remove one of the path, suppose e. Then, there is still an u-v path from u to v (P), which means that the vertices u,v are still connected
4. Since the contrapositive is true, then the original proposition must be true.

□

3 Trees

3.1 Overview

1. Tree Fundamentals and Properties

3.2 Fundamentals and Properties

Definition 3.2.1 (Trees)**.** *A tree is a connected acyclic graph.*

Definition 3.2.2 (Leaf)**.** *The end vertices in a tree is called a leaf. If v is a leaf, then deg(v)=1*

Remarque (Alternative Definitions to trees)**.**

1. *A graph is a tree* $\iff$ *a graph has n-1 edges*
2. *A graph is a tree* $\iff$ *every edge is a bridge*
3. *A graph is a tree* $\iff$ *Every adjacent vertices are connected by a unique path*

Problème (Show that a graph is a tree)**.** *To show a graph is a tree, we must show that*

1. *connectivity: all vertices are reachable from any vertices*
2. *acyclic: show that it can't have a cycle by contradiction*

Theorem 3.2.1 (A graph is a tree iif each distinct pair of vertices is connected by a unique path)**.**

1. $\Longrightarrow$*: Show that there cannot be two path by contradiction*
   - *Let T be a tree (no cycle+connected)*
   - *SLC, c-a-d qu'il existe 2 u-v path.*

- *However, if there exist two distincts uv path, then we have a cycle, which contradict the fact that T is a tree.*

2. $\Longleftarrow$ *: Use definition of tree*

   - *Let G be a graph where each pair of vertices has a unique path*
   - *Connectivity: if each pair has a unique path, then we can reach all vertices from anywhere*
   - *Acyclic: SLC, c-a-d que G has a cycle. This means that there exist two distincts uv-path from u to v. However, this contradict the fact that there exist a unique path from u to v.*
   - *Since G is connected and acyclic, then it is a tree*

Theorem 3.2.2 (Each nontrivial tree has at least two leaves)**.** *We can take the longest path and show that their end vertices must be deg(v)=1 by contradiction*

1. *Let T be a tree and P be the longest path* $P = (v_0 = u, ..., v_n = v), u \neq v$
2. *Consider the two end vertices u and v. We can show by contradiction that their degree must be 1*
   - *if u and v are connected to other vertices not in the path, then it wouldn't be the longest path, which contradict the fact that P is the longest path*
   - *If u and v were connected to other vertices in the path, then we would have a cycle, which contradict the fact that T is a tree*

Theorem 3.2.3 (Every tree of order n has size n-1)**.** *This theorem states that if a tree has n vertices, then it must have n-1 edges. It can be proven by induction*

1. *Cas de base: n=1*
2. *Suppose that the induction hypothesis is true ie that a tree with n vertices has n-1 edges. Show that a tree with n+1 vertices has n edges*
3. *Let T be a tree with n+1 vertices. Consider the tree T' in which we remove a leaf from T (if we remove a vertice other than a leaf, then it wouldn't be a tree). Therefore,* $deg(T') = deg(T) - 1 \Longleftrightarrow deg(T) = deg(T) + 1 = (n-1) + 1 = n$

Theorem 3.2.4 (Every tree is bipartite)**.** *We can use the fact that every bipartite graph has no odd cycle. Since a tree has no cycle, then the previous statement is true*

Remarque. *We could also make an argument that we can color all nodes without coloring two adjacent node from being the same color by making path.*

Theorem 3.2.5 (Connected Graph of order n has at least n-1 edges)**.** *TODO: Proof by minimum counter-example*

Remarque. *The previous theorem tells us implicitly that trees are minimally connected graph. In other word, the smallest connected graph we can form would be a tree*

Definition 3.2.3 (Forest)**.** *If all the component in a graph form a tree, then we say the graph is a forest.*

Theorem 3.2.6 (A forest has n-k edges)**.** *Let F be a forest of order n with k components. We can show that it has n-k edges directly*

1. *Let T_i be the component of the forest, where each tree has $n_i - 1$ edges (theorem: tree has n-1 edges)*
2. *Note that sum of vertices in all trees is the degree of the graph : $\sum T_i = n$*
3. *If we sum all edges in all trees, we get $\sum deg(T_i) - 1 = n - k$*

3.3 Minimum Spanning Tree

3.3.1 Overview

Un minimum spanning tree est un arbre dont le poids est minimal. Ce qu'on entend par un poids minimal, c'est que la somme totale de la distance entre chaque vertex soit minimale.

Theorem 3.3.1 (Every connected graph has a spanning tree)**.** *Proof by induction: G-e is still connected*

1. *Cas de base: 0 edges -¿ a vertex is a tree by default*
2. *Induction Hypothesis: Suppose every connected graph of n vertices has a spanning tree. Prove that a graph of n+1 vertices has a spanning tree.*
3. *Induction Step: Let G be a connected graph with n+1 vertices and let e be an edge of the graph. G-e is still connected (e is not a bridge since it removing it doesn't disconnect the graph).*

3.3.2 Algorithms to find a spanning Tree

Kruskal's Algorithm

Kruskal's Algorithm uses union-find to group similar vertices and connect these groups by the smallest edge

TODO

Prim's Algorithm

Prim's Algorithms works by choosing a random vertex and adding connecting the adjacent vertices that doesn't form a cycle with the current edges set.

TODO

Remarque. *Since we know a spanning tree has n-1 edges, we can iterate n-1 times instead of a while loop*

3.4 Non-Separable Graph: Vertex and Edge Connectivity

3.4.1 Overview

This section focusses on the relationship between vertices and connectivity. Specifically, we want to caracterize what makes a graph more connected than some others ones

1. Non-Separable graph
2. Cut Vertices
3. Vertex Connectivity
4. Edge Connectivity
5. Menger's Theorem

3.4.2 Non-Separable Graph

Definition 3.4.1 (Non-Separable Graph)**.** *A non-separable graph is a graph that is*

1. *nontrivial*
2. *connected*
3. *contains no cut vertices*

Theorem 3.4.1 (Caracterisation of a Non-sperable Graph)**.** *A graph with at least 3 vertices is non-separable iif two vertices in the graph lie on a common cycle*

Remarque. *On dit qu'un graphe doit avoir au moins 3 vertices, car "u—v" est aussi un cut vertex, mais n'a pas de cycle*

3.4.3 Cut Vertices

Definition 3.4.2 (Cut Vertices)**.** *A cut vertex is a vertex of a connected graph whose deletion makes the graph disconnected.*

Theorem 3.4.2 (Cut Vertices are $deg(v) \geq 2$)**.** *Let v be a vertex incident with a bridge in a connected graph G. Then, v is a cut vertex of G iif* $deg(v) \geq 2$

Proof. 1. $\Longrightarrow$: Proof by contrapositive

   - Let v be a vertex with deg(v)¡2 = 1. Therefore, v is a end vertices, so G-v is connected. Therefore, v is not a cut vertex

2. $\Longleftarrow$: Proof by contradiction

- Suppose $deg(v) \geq 2$ is not a cut vertex, and that G-v is connected. Then, uv is a bridge
- Since v is $deg(v) \geq 2$, then there is an adjacent edge let's say $vw \in E(G)$. Therefore, there is a u-w path
- However, the bridge lies on a cycle, and we know that a bridge can't be a cycle, so v must not be a bridge. However, this contradicts the fact that v is not a cut vertex. TO REVIEW

□

Theorem 3.4.3 (Caracterisation of Cut Vertices). *A vertex v of a connected graph G is a cut vertex iif there exist two vertices u and w distincts from v such that v lies on every u-w path in G-v*

Proof.

1. $\Longrightarrow$: Direct proof from the definition
   - Let G be a connected graph and v be a cut vertex. By the definition of a cut vertex, v must be a bridge and G-v must be disconnected. Therefore, $\nexists$ u-w path between G-v (but there is one in G)
   - Consequently, v must lie on every path, otherwise, there would be another path from u to w, which contradicts the fact that G-v is disconnected
2. $\Longleftarrow$: v lies on every u-w path -¿ v is a cut vertex
   - Let $v \in V(G)$ such that v lies on every u-w path.
   - Since G is nontrivial, then $u \neq v, v \neq w$.
   - Therefore, G-v is disconnected, which means that v is a bridge and that v is a cut vertex

□

Lemma 3.4.4 (Every non-trivial graph contains two vertices that are not cut vertices)**.**

Theorem 3.4.5 (The fartest vertex from another vertex is not a cut vertex)**.** *Let u be a vertex of a connected non-trivial graph G. Is v is a vertex fartest from u in G, then v is not a cut vertex.*

Proof. TODO □

3.4.4 Vertex Connectivity

Definition 3.4.3 (Vertex Cut)**.** *A vertex cut is a set of vertices u, that, when removed, disconnect the graph*

Definition 3.4.4 (Vertex Connectivity)**.** *The vertex connectivity of a connected graph is the minimum cardinality of all vertex cuts. Generally, we say it is the minimum number of vertices we can delete to disconnect G or to make it trivial. We say kappa of G* $K(G)$

Definition 3.4.5 (K-connected Graph)**.** *TODO*

Theorem 3.4.6 (Bound on vertex Connectivity)**.** *The minimum number of vertices needed to disconnect a graph or make it trivial is bounded by* $0 \leq K(G) \leq n-1$

Intuition. *Intuitivement,*

1. *The graph is already disconnected, we don't need to remove a vertex to make it disconnected -¿ 0 vertex*
2. *The graph is complete: we need to remove n-1 vertices to make it trivial*

Remarque (Show that graph is k-connected)**.** *To show that a graph is k-connected, we have to show that the lower bound is necessary and that the upper bound is sufficient.*

Proposition 3.1 (Petersen Graph and Vertex Transitivity)**.** *The Petersen Graph had a vertex connectivity of* $3 \leq K(P) \leq 3$*. Since removing a any vertex from the graph is isomorphic to any subgraph, we say the Petersen Graph is vertex transitive.*

3.4.5 Edge Connectivity

Definition 3.4.6 (Edge Cut)**.** *Similar to vertex cut, an edge cut is the set of edges X such that G-X is disconnected.*

Definition 3.4.7 (Edge Connectivity)**.** *The edge connectivity of a graph is the minimal of edge cut needed to make the graph disconnected. We write lambda* $\lambda(G)$

Remarque. $\lambda($ *Trivial Graph* $) = \lambda($ *Disconnected Graph* $) = 0$

Theorem 3.4.7 (Edge Connectivity of a complete graph is $\lambda(K_n) = n-1$)**.** *To show that* $\lambda(K_n) = n-1$*, we need to show that* $n-1 \leq \lambda(K_n) \leq n-1$ *ie that n-1 is necessary and sufficient*

3.5 Menger's Theorem

Definition 3.5.1 (Vertex Separating Sets)**.** *Let S be a vertex cut of G and let u and v be verteices fronc distincts components of G-S. Then S s a u-v separating sets. A minimum vertex separating set is the minimum of vertex we have to remove to disconnect vertices u and v.*

Remarque (A vertex separating set is not unique)**.**

Proposition 3.2. *If S is a u-v separating set, then* $|S| \geq K(G)$

Definition 3.5.2 (Vertex Disjoints Paths). *Let P_1 and P_2 be paths in the graph G. We say P_1 and P_2 are vertex disjoints path if these paths do not have vertex in common. We say P_1 and P_2 are internally disjoints if only their start and end vertices are the same*

Proposition 3.3.

1. *Example of disjoint path: $P_1 = a, b, c, d, P_2 = f, g, e$*
2. *Example of internal disjoint path: $P_1 = a, e, f, t, b, P_2 = a, g, h, b$*

Theorem 3.5.1 (If two paths are vertex disjoints, then they are edges disjoints)**.**

Intuition. *Si 2 chemins ne partagent pas les mêmes noeuds, alors ils ne peuvent pas avoir les mêmes arrêtes, car ils faut passer par les mêmes noeuds pour avoir les mêmes arrêtes*

Theorem 3.5.2 (Menger's Theorem). *Let u and v be non-adjacent vertices in a graph G. Then minimum number of vertices in a u-v separating set is equal to the maximum number of internally disjoint u-v path in G.*

Intuition. *"If we need to remove a minimum of 3 vertices, then there must be 3 internal disjoint paths." There shouldn't be less, otherwise removing the 3rd wouldn't make the u-v path disjoint, which contradict the definition of separating set. There shouldn't be less because there wouldn't be enough path to remove*

Remarque. *The Menger's Theorem tells us that is we know the minimum vertex separating set, then we know the cardinality of the internal disjoints sets without having to find them.*

Proof for Menger's Theorem. TODO □

4 Optimization

5 Shortest Path

6 Planar Graphs

6.1 Overview

1. What is a Planar Graph

7 Coloring

8 Flow

9 Ressources

9.1 Books

- Reinhard Diestel: Graph Theory - Discrete Structure by Michiel Sidt (recommended)

9.2 Courses

- Wrath of Math: Graph Theory Playlist - Sarada Herke: Graph Theory - Lecture Notes from JL Martin Math 105 - Topics in Mathematics: `https://jlmartin.ku.edu/courses/math105-F11/`

9.3 Exercices

- Introduction to Graph Theory by Douglas B. West: Proofs-based book on graph theory - Combinatorics and Graph Theory - Vasudev - Schaum's Outline Graph Theory