Math 158 Lecture Notes (Professor: Jacques Verstraete)

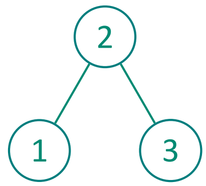
Isabelle Mills

January 17, 2024

Lecture 1: 1/9/2024

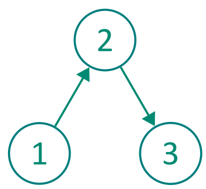
A graph is a pair (V, E) where V is a set of vertices and E is a set of unordered pairs of elements of V called edges. For $u, v \in V$, we say u and v are adjacent if $\{u, v\} \in E$.

For example: $G = (\{1, 2, 3\}, \{\{1, 2\}, \{2, 3\}\})$



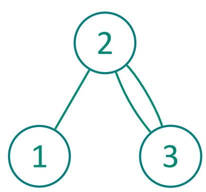
A directed graph (a.k.a a digraph) is a pair (V, E) where V is a set of vertices and E is a set of ordered pairs of elements of V.

For example: $G = (\{1, 2, 3\}, \{(1, 2), (2, 3)\})$



A multigraph is a pair (V, E) where V is a set of vertices and E is a multiset of unordered pairs of elements of V.

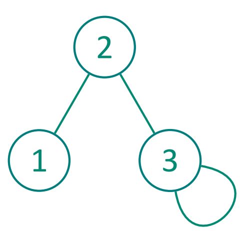
For example: $G = (\{1, 2, 3\}, \{\{1, 2\}, \{2, 3\}, \{2, 3\}\})$



A pseudograph is like a graph and multigraph except that the pairs in E are multisets.

Essentially, an element $\{a, a\}$ can belong to E in a pseudograph. This type of edge is called a loop.

For example: $G = (\{1, 2, 3\}, \{\{1, 2\}, \{2, 3\}, \{3, 3\}\})$



If $G = (V, E)$ and $v \in V$, the neighborhood of v is $N_G(v) = \{w \in V \mid \{v, w\} \in E\}$.

The degree of v is $d_G(v) = |N_G(v)|$. Or in other words, v's degree is equal to the number of edges connecting to v.

The Handshaking lemma states that for any graph (V, E):

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

The reason for this is that each edge increments the degrees of exactly two vertices. So the above sum counts every edge twice.

Lemma: Every graph has an even number of vertices with odd degrees.

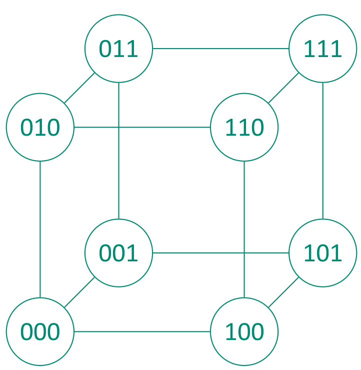
Proof: We can split the vertices of any graph into two categories: those with odd degrees, and those with even degrees.

Now recall that an even number plus an even number always equals an even number, as does an odd number plus an odd number. However, an odd number plus an even numbers equals an odd number. Based on this fact, we can guarentee that the sum of even degrees in any graph is even. And since the sum of even degrees plus the sum of odd degrees must be even as it equals $2|E|$ by the Handshaking lemma, we thus know that the sum of odd degrees must be even. Hence, it must be the case that there are an even number of vertices with odd degree because otherwise the sum of their degrees won't be even.

A graph is called r-regular if all of its vertices have degree r.

Note that the number of edges in any n-vertex r-regular graph is $\dfrac{rn}{2}$.

An r-dimensional cube graph, denoted as Q_r, is a graph such that $V(\mathrm{Q}_r)$, the set of vertices in Q_r, is equal to the set of binary strings of length r; and $E(\mathrm{Q}_r)$, the set of edges in Q_r, is equal to the set of pairs of binary strings which differ in only one position.



$$|V(\mathrm{Q}_r)| = 2^r$$

$$|E(\mathrm{Q}_r)| = \tfrac{2^r r}{2} = 2^{r-1} r$$

Note that Q_r is r-regular.

If $G = (V, E)$, then $H = (W, F)$ is a subgraph of G if $W \subseteq V$ and $F \subseteq E$.

If $W = V$, then H is a spanning subgraph of G (meaning that H has the same vertices as G but is lacking some of G's edges)

We define subtracting a set of vertices from a graph as follows:

For $G = (V, E)$ and $X \subset V$, we define...

$$G - X = (V \setminus X, \{\{u, v\} \in E \mid \{u, v\} \cap X = \emptyset\})$$

We define subtracting a set of edges from a graph as follows:

For $G = (V, E)$ and $L \subset E$, we define...

$$G - L = (V, E \setminus L)$$

Lecture 2: 1/11/2024

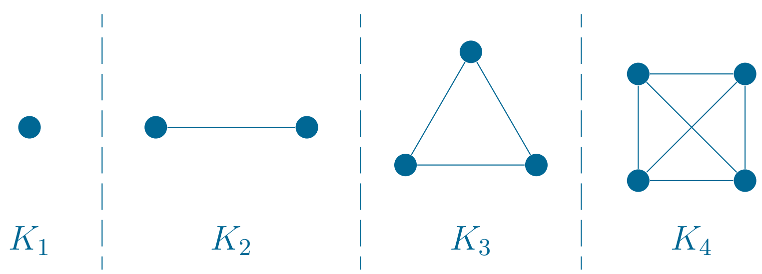
We shall notate that H is a subgraph of G by writing $H \subseteq G$.

An induced subgraph of $G = (V, E)$ is a subgraph $G[X] = G - (V \setminus X)$ where $X \subseteq V$. Alternatively, this is called the subgraph induced by X.

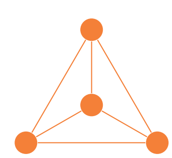
Given $G = (V, E)$ and $F \subseteq E$, the subgraph spanned by F is the subgraph whose edge set is F and whose vertex set is $\bigcup_{e \in F} e$.

Here are some basic classes of graphs:

- Complete graphs / cliques, denoted K_n, are graphs where every possible edge is present between n vertices.

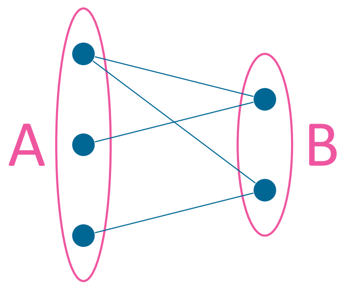


Note we can also draw K_4 such that there are no edge interceptions as follows:



$$|V(K_n)| = n$$
$$|E(K_n)| = \binom{n}{2} = \frac{n(n-1)}{2}$$

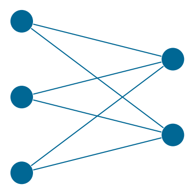
- A graph $G = (V, E)$ is bipartite if there exists a partition (A, B) of V such that every edge in E has one end in A and one end in B.



The partition (A, B) is called the bipartition of G. Then A and B are called the parts of G.

- A Complete bipartite graphs $K_{s,t}$, is the bipartite graph with parts A and B where $|A| = s$, $|B| = t$, and all possible edges between A and B exist.

For example, $K_{3,2}$:

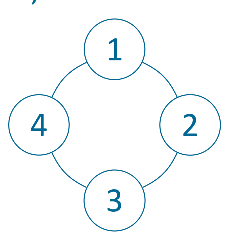


- A path P_k of length k has a vertex set $V = \{v_1, v_2, \ldots, v_k, v_{k+1}\}$ and an edge set $E = \{\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{k-1}, v_k\}, \{v_k, v_{k+1}\}\}$.

Note that $|V(P_k)| = k + 1$ and $|E(P_k)| = k$.
Therefore, below would be P_3...

- A cycle C_k of length k has a vertex set $V = \{v_1, v_2, \ldots, v_k\}$ and an edge set $E = \{\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{k-1}, v_k\}, \{v_k, v_1\}\}$.

Note that $|V(C_k)| = k$ and $|E(C_k)| = k$.
Therefore, below would be C_4...



Here is some terminology before the next lemma. For the graph $G = (V, E)$...

- $\delta(G) = \min\{d_G(v) \mid v \in V\}$ is the minimum degree of G.
- $\Delta(G) = \max\{d_G(v) \mid v \in V\}$ is the maximum degree of G.
- The degree sequence of G is the sequence of degrees of vertices G in non-increasing order.

Lemma (part 1): If $G = (V, E)$ is a graph of minimum degree $k \geq 2$, then G contains a cycle of length at least $k + 1$.

Proof: Let P be a longest possible path in G, say:

$$V(P) = \{v_1, v_2, \ldots, v_r\}$$

Then $N(v_r) \subseteq V(P)$. After all, if this were not the case, we'd be able to extend the path to the vertex in $N(v_r)$ but not in $V(P)$, thus contradicting the fact that P is a longest path.

Let v_i be the first neighbor of v_r along the path from v_1 to v_r. Then $\{v_i, v_{i+1}, \ldots, v_r\}$ are the vertices of a cycle C.

Now note that because $N(v_r) \subseteq P$ and v_i was the first element in the path P to belong to $N(v_r)$, we know that C contains all the elements of P that $N(v_r)$ also has. So, $N(v_r) \subseteq C$.

But now note that $|N(v_r)| \geq \delta(G) = k$. Plus, v_r itself is not in $N(v_r)$. Combining these facts together, we can say that the cycle C has at least $k + 1$ vertices.

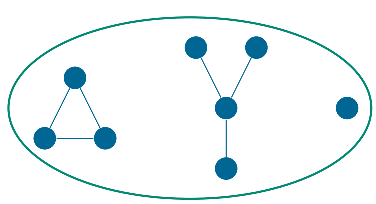
Lemma (part 2): The cycle length $k + 1$ is the longest we can guarentee based on the minimum degree of the graph being k.

Proof: Take the graph K_{k+1} which has a minimum degree k. Obviously, the longest cycle in K_{k+1} is the cycle containing all $k + 1$ elements of K_{k+1}. Thus, we have shown that there are graphs with minimum degree k which don't have cycles of length greater than $k + 1$.

A connected graph is a graph in which any two vertices are the ends of a path.

The components of a graph are the maximal connected subgraphs. For example:

Let us define G as:



As can be seen, G has three components.

A tree is a connected graph with no cycles (a.k.a it is acyclic).

Some examples of small trees include: K_1, K_2, $K_{1,2}$, P_3, and $K_{1,3}$.

Lemma: Every tree with n vertices has exactly $n-1$ edges.

Proof: We shall proceed by induction.

If $n=1$, the tree is K_1, meaning that it has $0=n-1$ edges.

Now assume the lemma is true for all trees with n vertices, and let T be a tree with $n+1$ vertices. Then, we shall remove a vertex v of T with degree 1. (Note that we know such a vertex must exist since otherwise the minimum degree of T would be at least 2 and that would guarentee a cycle exists of at least length 3. This of course contradicts the fact that T is acyclic.)

Then $T-\{v\}$ is a tree with n vertices as it must be acyclic and connected. So by induction it has $n-1$ edges. And because v has degree 1, we know that
$|E(T)| = 1 + |E(T-\{v\})| = 1 + (n-1) = n$.

Lemma: Any connected graph with finite vertices has a spanning tree.

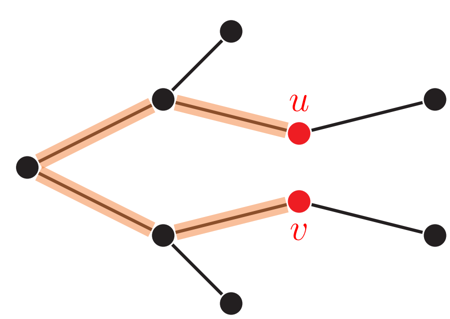
Proof:

Firstly consider the case that the graph G has no cycle. Then, it is a tree by definition.

Now, consider if G has a cycle C. Then for any edge $e \in E(C)$, we have that $G-\{e\}$ is still connected. So, we can now go back to the top of the proof and ask: does $G-\{e\}$ have any cycles? We can repeatedly do this until the graph has no cycles since taking away edges does not remove any vertices.

This actually acts as an algorithm for finding a spanning tree of any connected graph.

If u and v are two vertices in a connected graph, the distance from u to v is the length of a shortest path with ends at u and v.



Let $d_G(u, v)$ be the distance between u and v.

Distance is a metric, meaning:

1. $d_G(u, v) = 0 \iff u = v$
2. $d_G(u, v) = d_G(v, u)$
3. $\forall w \in V, \;\; d_G(u, v) \leq d_G(u, w) + d_G(w, v)$

The diameter of a connected graph G is the maximum distance between any two vertices of G. Or in other words, $\max\{d_G(u, v) \mid u, v \in V(G)\}$.

The radius of G is equal to $\min\{\max\{d_g(u, v) \mid u \in V(G)\} \mid v \in V(G)\}$. What that means is that the radius of G measures the smallest distance path one could limit themselves to drawing while still being able to have that path have one end at some fixed vertex and its other end at any arbitrary vertex in the graph.

Examples:

1. The radius of K_n is 1. The diameter of K_n is n.

2. The diameter of P_k is k. The radius, can be computed as follows:

   The middle vertex of a path will have the fastest access to either end of the path. So, we shall measure the radius from the vertex: $v_{\lceil \frac{k+1}{2} \rceil}$. Then, we can see that v_{k+1} is going to be a farthest element from $v_{\lceil \frac{k+1}{2} \rceil}$. So the radius of P_k equals $k + \lceil \frac{k+1}{2} \rceil$.

   Now you can consider what happens when k is even and odd. But what's important is that it works out that the radius is $\lceil \frac{k}{2} \rceil$.

We can use a search tree to more generally find the radii and diameters of graphs.

Breadth-First-Search

Here's how to find a spanning tree in a connected graph with a root vertex v such that the tree "preserves" all distances from v. (This tree is called a BFS tree).

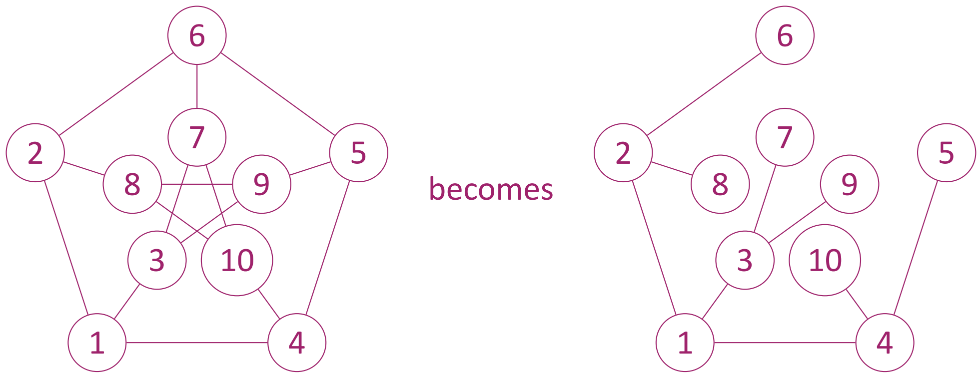
Let G be a connected graph and let $(v_1, v_2, v_3, \ldots, v_n)$ be any ordering of the vertices of G.

Pick a vertex $v = v_1$ to be the root of the BFS tree.

Now, at any stage in constructing this tree, we will have a vertex set $V(T) = \{v_1, v_2 \ldots, v_k\}$ (when we first start, $V(T)$ will only contain v_0. So don't worry about that). Now if $V(T) = V(G)$ we can stop. Otherwise though, we can say that there is a smallest integer i such that for $v_i \in V(T), \;\; N(v_i) \setminus V(T) \neq \emptyset$. Choose v_{k+1} to be the smallest neighbor (by the ordering of $V(G)$) of v_i not in T and add the edge $\{v_i, v_{k+1}\}$ to T. Then we repeat this paragraph.

Beware the ordering we are creating in our tree will often be different from the order of the graph you started with.

Worked Demonstration:



Properties of BFS:

- If the root is v, then $d_T(v, w) = d_G(v, w)$. In otherwords, a BFS tree preserves distances from its root.
- The Tree with root v has layers $N_i(v) = \{w \in V(G) \mid d_G(v, w) = i\}$. Furthermore all edges in the original graph either stay inside a single layer $N_i(v)$ or go between adjacent layers (i.e. from $N_i(v)$ to $N_{i+1}(v)$).

  If an edge did "jump over" a layer, that violate the fact that distance is a metric.
- The diameter of G equals the maximum number of layers of all BFS trees (not including the 0-layer).
- The radius of G equals the minimum number of layers of all BFS trees (also not including the 0-layer).

Lecture 3: 1/16/2024

Note that a tree is "minimally connecting" as subtracting any edge from a tree will produce a disconnected graph.

We know this is the case because if we could remove an edge and still have the graph be connected, then that would imply the existence of a path between two neighboring vertices that doesn't go through their shared edge. But then, we'd be able to make a cycle subgraph by adding their shared edge to that path.

Depth-First-Search

Here is alternate algorithm for generating a spanning tree of a connected graph. A resulting tree of this algorithm is called a DFS tree.

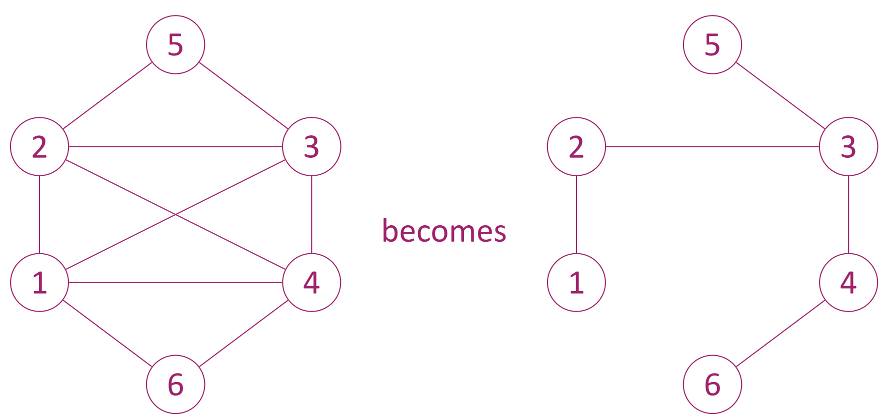
Let G be a connected graph and let $(v_1, v_2, \ldots, v_n)$ be any ordering of the vertices of G.

Pick a vertex $v = v_1$ to be the root of the DFS-tree.

Now, at any stage in constructing this tree, we will have a vertex set $V(T) = \{v_1, v_2, \ldots, v_k\}$. If $V(T) = V(G)$, we can stop. Otherwise though, we select i to be the largest integer such that for $v_i \in V(T)$, $N(v_i) \setminus V(T) \neq \emptyset$. Then, choose v_{k+1} to be the smallest neighbor (by the ordering of $V(G)$) of v_i not in $V(T)$ and add the edge $\{v_i, v_{k+1}\}$ to T. Then we repeat this paragraph.

Once again beware the ordering we are creating in our tree will typically be different from the order of the graph you started with.

Worked Demonstration:



Theorem: A graph is bipartite if and only if it contains no odd cycles.

Proof:

($\Longrightarrow$) First note that an odd cycle isn't bipartite. Thus, any graph containing an odd cycle is not bipartite.

($\Longleftarrow$) Now supposed we are given some graph G with no odd cycles.
Then, assuming G is connected (if G isn't connected, we can break G up into its component subgraphs and do this process for each component), we can construct a BFS-tree in G rooted at some $v \in V(G)$. Let us name this tree T.

Now as noted before, T will have layers L_i where each $L_i = \{u \in V(G) \mid d_G(v, u) = i\}$. Using those layers, we can partition T into two subsets A and B where A is the union of all L_i where i is even and B is the union of all L_j where j is odd. So, T is clearly bipartite.

Now, let's reinsert the removed edges from G back into T. Note that for each re-inserted edge e, it must be the case that either e is a subset of some L_i or that e goes between some L_i and L_{i+1}. Importantly, edges of the latter case do not violate our partition. So, if all the edges in $E(G) \setminus E(T)$ go between layers, then we can conclude that G is definitely bipartite just like T.

With that, we now intend to show that an edge G having an edge belonging to a single layer L_i guarentees that G contains an odd cycle.

Assume the graph G has an edge $\{u, w\} \subseteq L_i$ where L_i is the ith layer of a BFS tree rooted at v. Then, we know that there exists a path P_1 contained in that BFS tree going from v to u and a path P_2 contained in that BFS tree going from v to w. In order to draw a cycle from this information, let x be the vertex of some L_j such that $x \in V(P_1)$, $x \in V(P_2)$, and j is as large as possible. That way, by defining the subpaths ${P_1}'$ going from x to u and ${P_2}'$ going from x to w, we can get the following cyclic subgraph of G :

$$C = (V({P_1}') \cup V({P_2}'), E({P_1}') \cup E({P_2}') \cup \{u, w\})$$

However, now note that $|E({P_1}')| = |E({P_2}')| = i - j$. Hence, $|E(C)| = 2(i - j) + 1$, which in turn means that C has an odd number of edges. So, we have shown that if a graph G contains an edge within a single layer L_i, then we can give an example of an odd cycle within G.

So in conclusion, if we assume G has no odd cycles, then G can't have any edges which are subsets of a single layer L_i. But that means that every edge in G respects the partition we made to show that T is bipartite. So, G must also be bipartite with the same partition as T. ■

A Hamiltonian cycle is a spanning cycle of a graph. We say a graph is Hamiltonian if it contains such a cycle.

A walk is a sequence of vertices and edges: i.e. $(v_1, \{v_1, v_2\}, v_2, \{v_2, v_3\}, \ldots)$

Note that a walk can go over the same edge or vertex multiple times.

A trail is a walk with no repeated edge.

Interestingly, all paths are trails and all trails are walks. So a trail is kind of a middle concept between being a walk or a path.

A tour is a trail with the same first and last vertex.

So, all cycles are tours and all tours are walks.

An Eulerean tour of a graph is a tour which contains all the edges of the graph. If a graph has an Eulerean tour, we say it is Eulerean.

> For context, the name Eulerean comes from the fact that a mathametican named Euler from the 1700s asked how one could tell if a graph is Eulerean or not (look up the Seven Bridges of Königsburg problem).

Lecture 4: 1/18/2024