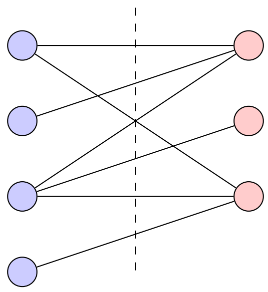
Graph Theory

Last updated: Thursday 17th March, 2016 12:51

Let us consider the problem of whether, on average, male has more female facebook friends or female has more average male friends on facebook. This seems to be a good measure of how each sex is more popular than the other.

The relationship between male and female on facebook can be represented with nodes representing users and the line connecting the users represent whether they are friends or not.



So, if we need to know the average number of female friends for a male user, then all we need to do is find total number of female friend male users has. This is just the number of edges(lines). So the average of opposite is

$$\text{Male Popularity} = \frac{\#\text{ of edges}}{\#\text{ of male}}$$

Similarly, the average male friend for female users is given by

$$\text{Female Popularity} = \frac{\#\text{ of edges}}{\#\text{ of female}}$$

The ratio of the two, in fact, has nothing about how popular each sex is on average. It just has to do with the total number of each sex in the population.

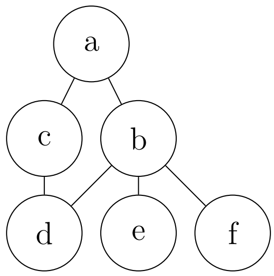
$$\frac{\text{Male Popularity}}{\text{Female Popularity}} = \frac{\#\text{ of female}}{\#\text{ of male}}$$

Graph

In the former example, we can see that the problem becomes much clearer if we strat drawing doodles representing objects of interest and the relation among them. Since the doodle has so many uses it has a fancy name called graph.

Def: A simple graph G consists of non-empty set V, called vertices or nodes of G and a set E of two element subset of V. The member of E are called edges of G. We write $G = (V, E)$.

For example, the following graph



The vertex are

$$V = \{a, b, c, d, e\}.$$

The edges are

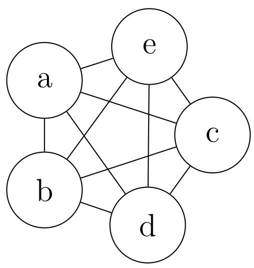
$$E = \{(a, b), (a, c), (c, d), (b, d), (b, e), (b, f)\}.$$

Def: Two vertices of in a simple graph are adjacent if there is an edge connect the two.

Def: An edge is said to be incident to the vetices it joins.

Def: The number of edges incident to the node is called the degree of the vertex.

For example, a complete graph of n vertices has $\binom{n}{2}$ *edges*. The figure below shows a complete graph of 5 vertices, K_5.



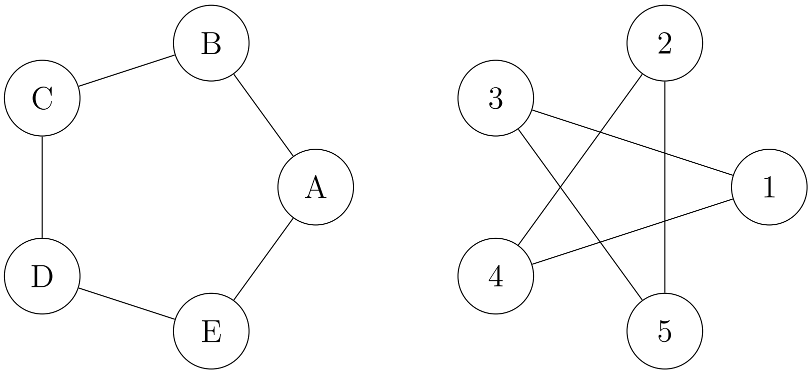
Isomorphism

Def: If $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ then G_1 is isomorphic to G_2 iff there exists a bijection $f : V_1 \to V_2$ such that for all pair $u, v \in V_1$

$$\{u, v\} \in E_1 \text{ iff } \{f(u), f(v)\} \in E_2$$

The bjection f is called an isomorphism between G_1 and G_2.

In simple words, G_1 and G_2 are isomorphic if they are the same graph up to relabeling. For example, the two graphs below are isomorphic.



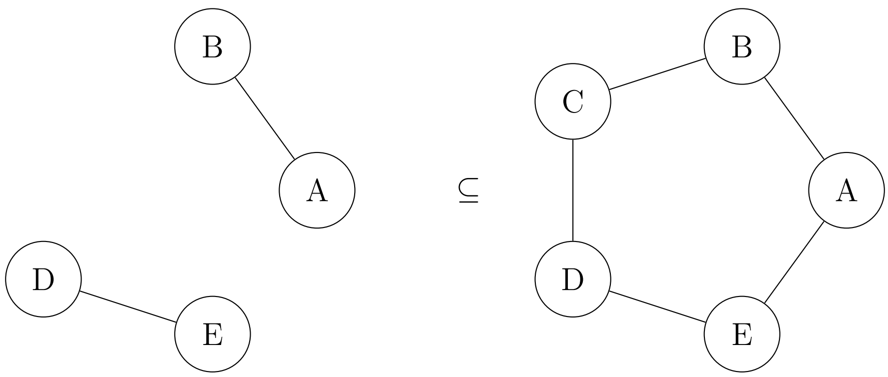
The isomorphism is

$$1 \to A, 3 \to B, 5 \to C, 2 \to D, 4 \to E$$

Subgraph

Def: A graph $G_1 = (V_1, E_1)$ is a subgraph of $G_2 = (V_2, E_2)$ if $V_1 \subseteq V_2$ and $E_1 \subseteq E_2$. We write $G_1 \subseteq G_2$.

For example,



Handshaking Lemma

Lemma: The sum of degree is $2|E|$.

Proof: Every edge contribute two to the sum of the degree. $\square$

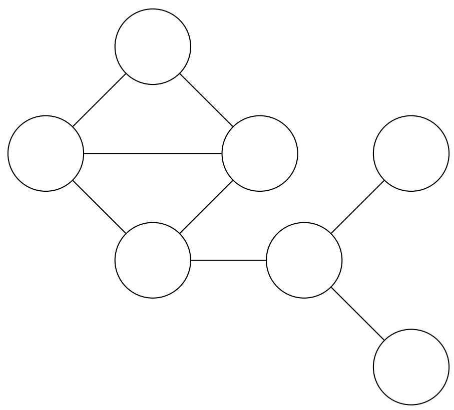
Collorary: In any graph there are even number of vertices of odd degree.

Proof: Otherwise the total number of degree would be odd. $\square$

Coloring

Let us consider the problem of assigning exam to timeslot. We do not want to have two exams on the same time slot.

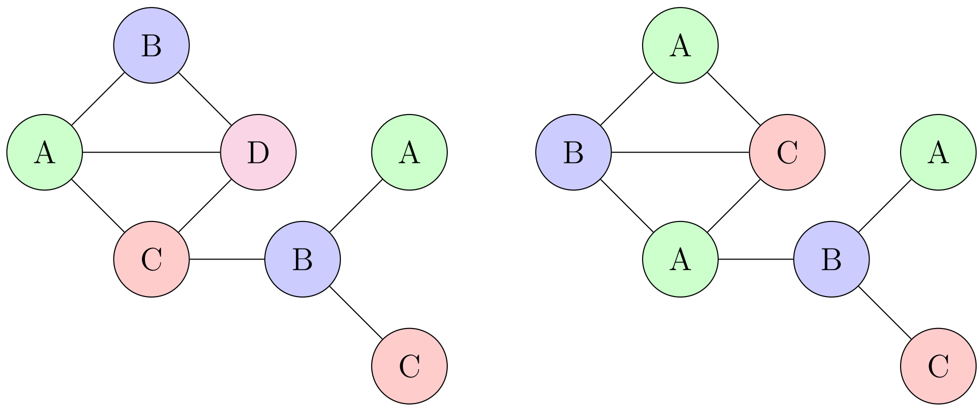
Let us draw a graph for this. Let the node represent the class and the edge represent whether there is any student taking the two classes. Let suppose the graph for our classes look like this



Suppose that you have couple time slot of the exams, A:8-10am, B:10-12pm, C:12-2pm, D:2-4pm, E:4-6pm.

One obvious solution is to give each of the class their own time slot. But, no one wants to take exam in the late evening. So we want to use as least timeslots as possible. On the other hand, if we don't have enough time slot there will be time conflict between some exams.

The problem we have here is to assign the timeslot(color) to the class(node) such that no two adjacent node has the same color. This is called coloring problem. Here are some possible assignments.



Both of the graph above are valid coloring scheme. But, they do not use the same number of color. The minimum number of color needed to color a graph is called the chromatic number.

Def: The minimum number of color needed to color a graph G is called chromatic number. $\chi(G)$

Trying to figure out the chromatic number for a gerneral graph is an NP complete problem (which means the best way to solve it as far as we know is non polynomial time.).

That being said it is quite easy to find a valid coloring (not necessarily optimal) scheme. Let us consider a greedy algorithm.

- First, we order the vertices (doesn't matter how).

$$v_1, v_2, v_3, \ldots, v_n$$

- We then order the color

$$c_1, c_2, c_3, \ldots, c_m$$

- Go through each vertex in the list give it the lowest color.

We can also bound the

Theorem: A graph with a maximum of d degree is $d+1$ colorable.

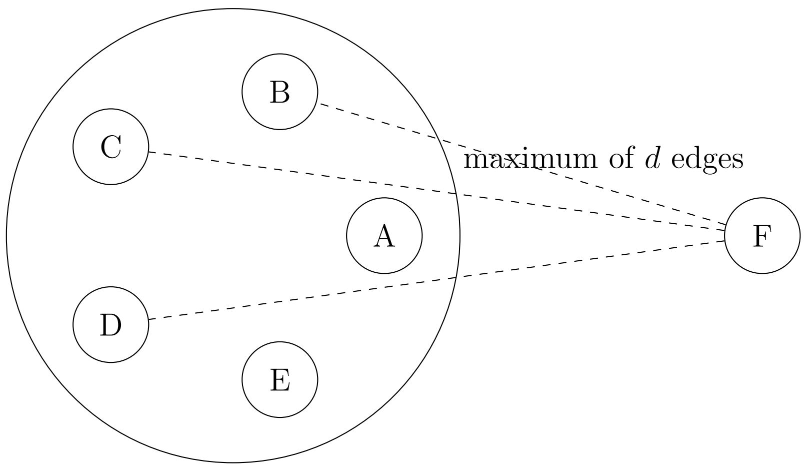
Proof: We will do this by induction. Miss it? For induction on graph always try to induct on number of node or edges. Never ever do induction on the number of degree.

Inductive Predicate: $P(s)$:= Graph of s nodes with maximum degree of d is $d+1$ colorable for all positive integer d.

Base Case: Graph of 1 node has 0 degree and is 1 colorable.

Inductive Step: The strategy here is to take away one node so we have a smaller graph we will then add it back.

- Let us assume that any graph of k node with maximum degree of d is $d+1$ colorable for all d.
- Let us consider a graph of $k+1$ node with maximum degree of d.

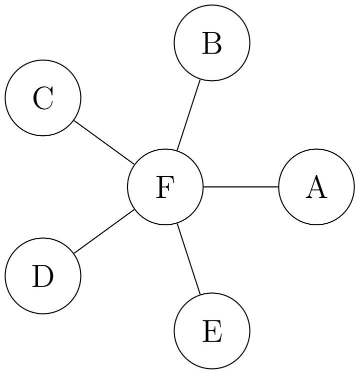


k nodes with maximum of d degree.

- We can split it into
  - a.) A subgraph of k nodes and the maximum degree d. By inductive hypothesis, we know that this subgraph need at most $d+1$ color.
  - b.) A node F.
- Since the degree of F is at most d, the number of edge linking F back to the subgraph is at most d.
- The subgraph use $d+1$ color and there are at most d edges link F with the subgraph. There is at least 1 color from $d+1$ color that is not used by any node connected to F.
- We can use that color to color F. We do not need extra color for F. So, the total color needed is $d+1$ for the subgraph.

$\square$

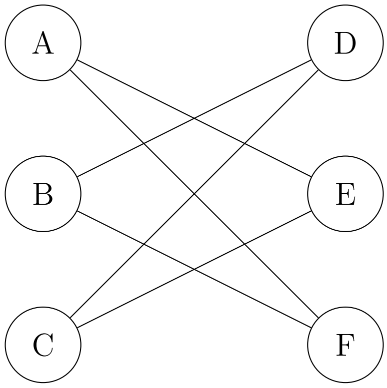
It should be noted that this may give you an unnecessary large upper bound. Let us consider a graph that looks like the following



Even though you only need two color for the above graph. The bound we proves says that we need 6 color to do it.

When the Algorithm does not do well

There are certain case where the algorithm does not do too well. If we have a graph that looks like the following:



If we order the vertices such that we order it like

$$A, D, B, E, C, D$$

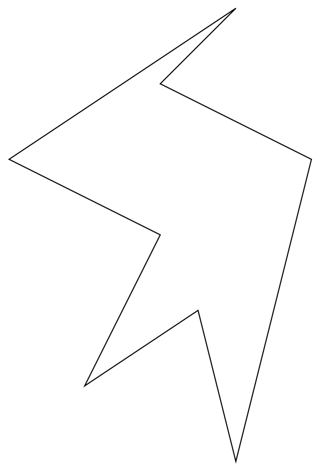
then after you assign c_1 to A then you will also assign c_1 to D. c_2 will be assigned to B and E. Then c_3 will be assign to C and F and so on. We will need $|V|/2$ color instead of just 2 colors.

Cell Tower Placement

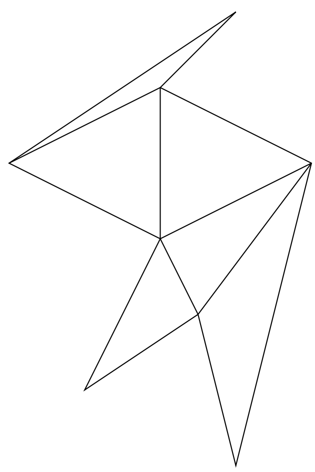
Coloring problem is far more reaching than just mathematical play toy. Let us consider the problem of placing cell tower. Each cell tower has some range. But there is an annoying physics part that if two cell tower are in range of each other, we cannot use the same frequency since they will interfere. So, for the two cell towers that has overlapping area we need to assign different frequencies to each tower.

Art Museum Theorem

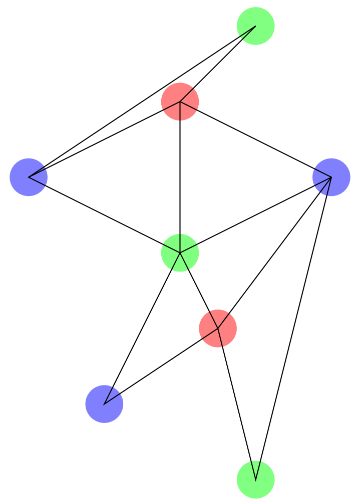
Let us consider the problem of guarding an art museum. Let us assume that art museum is a simple polygon(no hole). We need to place guards such that we cover the whole museum. One way to do it is to place the guard at every single corners but that would be overkill. The cost for security guard will run too high. So we need to figure out the way to find smaller number of guards.



Let us triangulate the museum. The idea is that to cover a triangle all we need to do is place a guard at one of the corner then the triangle is covered. We will prove later that triangulation is always possible.



That means all we need to do now is to color the nodes. Each color represent each set of different placement of guard. We don't want the same set of guard to be guarding the same triangle since that would be a waste of money. To do this we only need 3 colors.



For the example above we have 2 red, 3 blue and 3 green. That means if we place the guards at 3 blue verticecs we will cover all the regions. Or we can place just 2 guards at the 2 red vertices then we have every thing covered. Let n be the number of vertices and r be number of red vertex, b for the number of blue vertex, and g for the number of green vertex. Then we have

$$r + g + b = n$$

This means that there is at least one of the color with less or equal to $n/3$. Otherwise the sum will be greater than n.

The result is called Art Muesum Theorem saying the minimum number of guard needed to guard n vertex simple polygon is at most $n/3$. This elegant prove is by Steve Frisk while dozing off a bus in Afganistan.[1]

Now let us fill in the missing holes.

Triagulation is Always Possible

Theorem: Given a polygon triangulation is always possible.

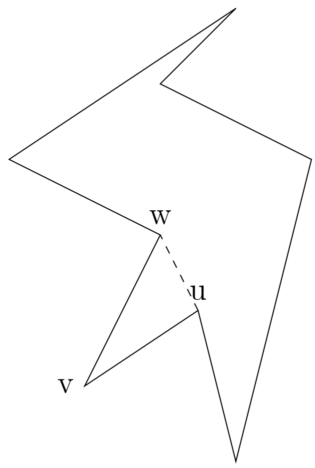
Proof: We will show this by induction on the number of vertex. You can fill in the formality.

[1] http://bowdoinorient.com/article/4980

If we have a triangle then it is trivial to triangulate. It is already a triangle. There is nothing to do here.

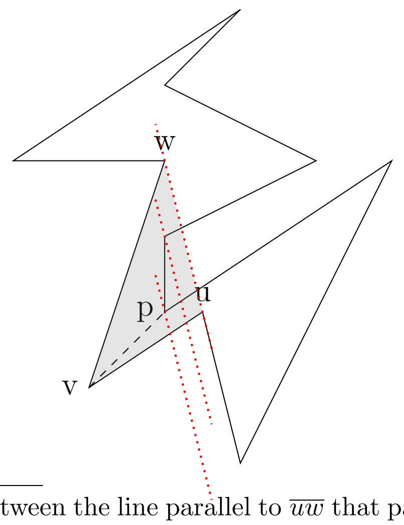
Let us consider an n vertex polygon. The consider three adjacent vertices v, u and w. There are two cases for the three vertices.

a.) If the triangle form by the three vertices has no node inside then we have a triangle and an $n-1$ vertices polygon. We know we can triangulate both by inductive hypothesis.



b.) If the triangle formed by u, v and w has some vertices inside. Then, all we need to do is pick the vertex that is closest[2] to v that is not u or w. let us call that point p.

The reason that we pick such a line is because the line from v to p cut the polygon in to two smaller polygons[3] which we know can be triangulated.



[2]closest here means the distance between the line parallel to $\overline{uw}$ that pass through that point is closets to v

[3]If it has a hole then all we need to do is keep cutting it if it doesn't split. The number of hole is finite and each time if we don't split the polygon we will destroy a hole. We guarantee to destroy all holes.

□

Two ears theorem

Theorem: Every simplex polygon with $n \geq 4$ has at least two ears. An ear is a three adjacent vertices that form a diagonal line(no other vertex in the triangle).

Proof: We can proof by the same construct as the previous problem by picking three vertices.

The base case is a square. That's trivial.

Let us consider simple polygon with $n > 5$

a.) If u, v and w form an ear, then we end up with an ear and a smaller polygon with 2 ears by IH. When we combine them back since they share at most 1 edge(no hole), we can destroy at most 1 ear if that edge belongs to an ear of the left over polygon. So the number of left over ear is still ≥ 2.

b.) If u, v and w does not form an ear, we again pick a point p "closest" to the v and cut the polygon into two. There are two cases.

  a.) One of them is triangle. Then we are back the previous case. Adding a triangle reduce the nubmer of edge.

  b.) If none of it is a triangle. Then you get two from both sides and you destroy at most two for the edge that we share.

  c.) Ending up with 2 triangles is not possible since $n \geq 5$

3 Color Theorem

Theorem: Every triangulated graph of a simple polygon is 3 colorable. (No hole)

Proof: We will show this by induction on the number of triangles.

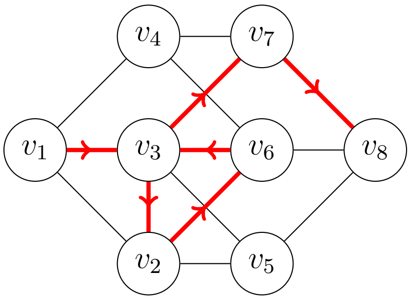
The graph with 1 triangle is trivial to do.

Let us consider a graph of n triangles and let us remove one ear. The graph without that ear is 3 colorable by IH. Then all we need to do at add an ear back is to add a vertex. Since this vertex has exactly 2 edge. We can color this vertex with the left over color. □

Path Walk Connectivity

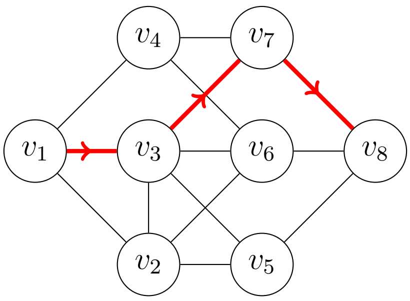
Def: A walk is a sequence of vertices that are connected by edges. For example, the path

$$v_1, v_3, v_2, v_6, v_3, v_7, v_8$$



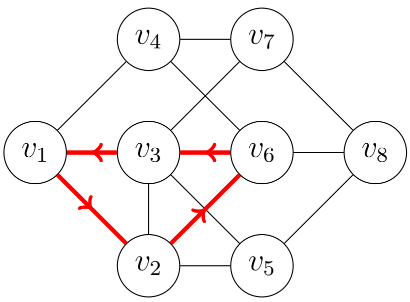
Def: A path is a walk where all the vertices are different. So the walk we gave in the earlier example is not a path. Here is an example of a path

$$v_1, v_3, v_7, v_8$$

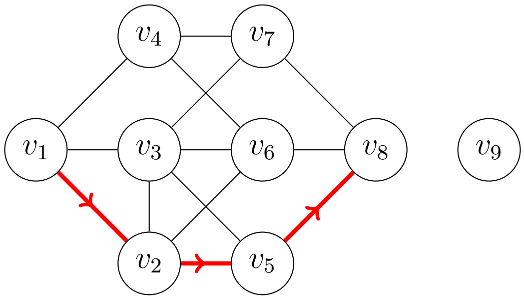


Def: A cycle is a walk that start and end at the same point. Here is an example of a cycle

$$v_1, v_2, v_6, v_3, v_1$$



Def: u and v are connected if there is a path from u to v. For example, v_1 and v_8 are connected. But, v_1 and v_9 are not connected. Moreover, a graph is connected if all pair of vertices are connected.



You may notice that we use the word path here but not a walk this is because of this little fact.

Lemma: If there is a walk from u to v then there is a path from u to v.

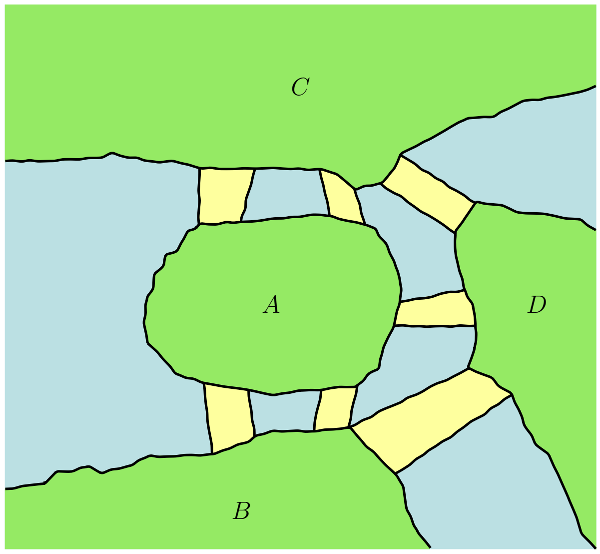
Proof: If the walk have a cycle then keep removing it. Then we will end up with a walk that have no cycle and that's a path. $\square$

Euler Walk

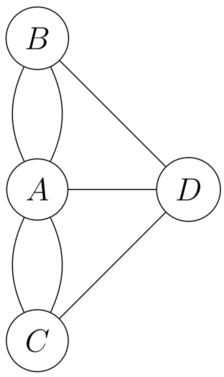
Now let us get to the fun path. There are so many thing we can do with it. Let us first consider a poor tourist problem.

The tourist which we will state the name later. Went to a little town in Germany(back then) called Konigsberg now in Russia. The town has 7 bridges and he wants to walk along all the bridge exactly one each. The map of the town is shown below[4].

[4] Courtesy from Stackoverflow `http://tex.stackexchange.com/questions/183882`



The poor tourist was actually Euler. The Euler. He turned this problem into a graph theory problem which many claim that this was the first graph theory problem. He turned this map into a graph[5].



So the problem of trying to walk along the bridge exactly once each is turned into the question whether there is a walk in the graph such that every edge is used exactly once.

Euler, as always, noticed something insightful about the graph.

- If you have an odd degree node, we will have to start our walk or end our walk there. This is because we have to use all the incidenting edge. We have to use it for going in and going out alternatively. That means if you have an odd degree node then, if you start with out and end with out or start with in and end with in. The first case means that node is the starting node and the second case means that this node is the end node.

[5]Not a simple graph since it has multiedge. But it is a graph.

The graph above has 4 odd degree vertex. But, we can only have one starting point and 1 end point. So, such a walk is not possible. This is extremely profound. So, we name this kind of special after him.

Def: An Euler Walk is a closed walk where all the edge are used exactly one.

Lemma: If a graph have more than two nodes of odd degree, then there is no Euler walk.

Proof: Done. □

Lemma: If you have one node of odd degree, you can not do it either.

Proof: If you start there you have to end somewhere. □

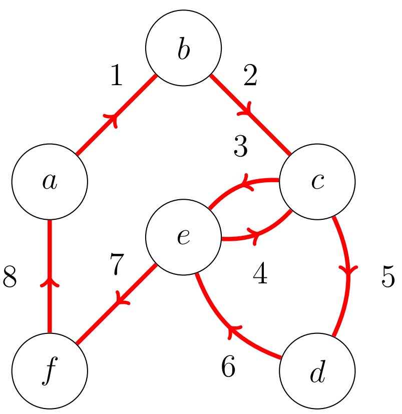
.

Lemma: If graph G is connected and 0 odd degree node(all degree are even) then it has an euler walk.

Proof: Let us first consider walks in G, such that it uses every edges at most once.

- We know that such walk exists since you have single node walk v_1 that use every edge at most once. Every edge is used exactly 0 time.
- Let us consider the longest of such walk let us call the walk w

$$w = v_1, v_2, v_3 \dots v_k \underset{\substack{\uparrow \\ \text{For example}}}{=} a, b, c, e, c, d, f, a$$



We know that it exists because it it gets too long then you will use an edge more than once. Recall pigeon hole principle. We hope that this walk will become an Euler tour. This means we want to make sure that

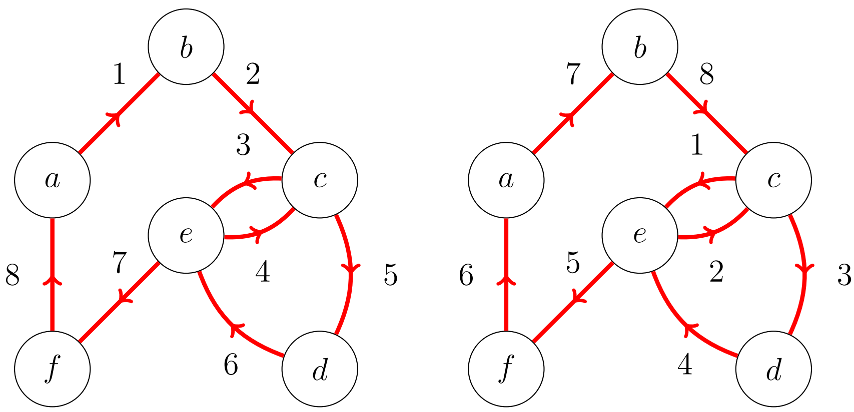
a.) It is a closed walk.

b.) No edge left behind.

- First we know that every edge connected to v_k must be used otherwise we will just extend the walk with that edge.
- This means that v_k must be the same as v_1 otherwise v_k have an odd degree. For even degree we need to start with going in then end with going out.
- This means v_k and v_1 are the same thing. So, w is a closed walk.
- So, all that is left to do is to show that every edge is used. This is also simple. We can rotate the closed walk we have around such that the node we want is at the end and use the same argument that every edge connected to that vertex is used. Since this is true for all vertices, all the edges must be used.

  For example, if we want to show that all edge incidenting to c is used all we need to do is consider w' which is just w rotated. We just shift everything by 2.

$$w = a, b, c, e, c, d, f, a \to w' = c, e, c, d, f, a, b, c$$



  Then we know that all the edge of c must be used otherwise c has an odd degree. We can do this for every vertex. So all the edge must be used.

- Thus, w is an euler tour.

$\square$

Collorary: If it has exactly 2 odd degree node then there is an euler walk. Moreover it has to start and end at the two odd degree nodes.

Proof: Let the two odd degree nodes be a and b. Then all we need to do is to add a helper edge incidenting to a and b. Then every node have even degree. So there is an euler tour. Then all we need is just remove that edge then the left over walk is an euler walk by contruction. □

Theorem: Euler's theorem. Euler walk exists if and only if the graph is connected and there are exatly 0 or 2 odd degree vertex. Morover, if there are 2 odd degree vertices then, the Euler walk must start and end at the 2 odd degree vertices.

Proof: All the above combine. □

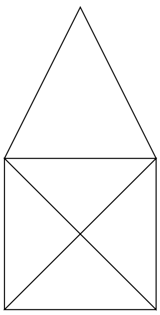
.

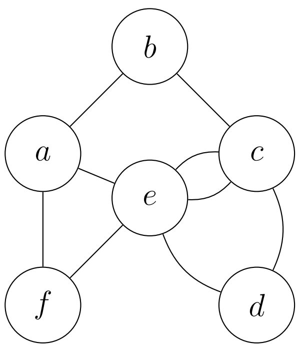
Drawing with One Stroke.

One direct application of this is a game we all have seen before. Draw something with one stroke. Or show that you can't draw it. This is exactly the problem of Euler's theorem. We learn couple things from it.

- If it doesn't have 0 nor 2 odd degree node, don't even try. Your friends are just trolling on you.
- If it has 2 odd degree vertices then, start at one and try to end at another one. You will see that it is quite easy if you know where to start and where to end.
- If it has 0 odd degree vertex then, start anywhere, and try to end at the same point.

There is actually an algorithm to find Euler Walk. You can look up Fluery's algorithm or Hierholzer's algorithm. But, if the graph is smaller just do trial and error. It is very likely that you will get it the first time. Here are some figures you can try:





DNA Chip

In the future, DNA sequencing will become one the most important medical diagnosis[6]. One of the most important advance in making DNA sequencing to economically feasible is the development of shotgun sequencing technique.

Here is our problem: we have a DNA sqeuence. It is a very long series of different types of nucelotides.

$$GATTACA$$

The old way to do it is called Sanger's method. (Look up youtube for it). Basically, it tries to terminate a copy of DNA at certain type of marked nucleotide. We can then sort the DNA by length and then see at what length correspond to which marked nucleotide It doesn't work well with long DNA chain. Though.

A shotgun sequencing is try to find all the possible length k substring of DNA chain. For example, the length 3 of the $GATTACA$ DNA chain above is

$$\{GAT, ATT, TTA, TAC, ACA\}$$

Short sequence is easy and cheap to detect. The device is called DNA microarray. You put DNA in follow the instruction and you will get what are the subsequence in your DNA.

But the problem is actually in reassembling it the fragment into the sequence. One thing we can do is try to find the shortest string that contains all these substring. This is sort of saying that such sequence is the most likely one.

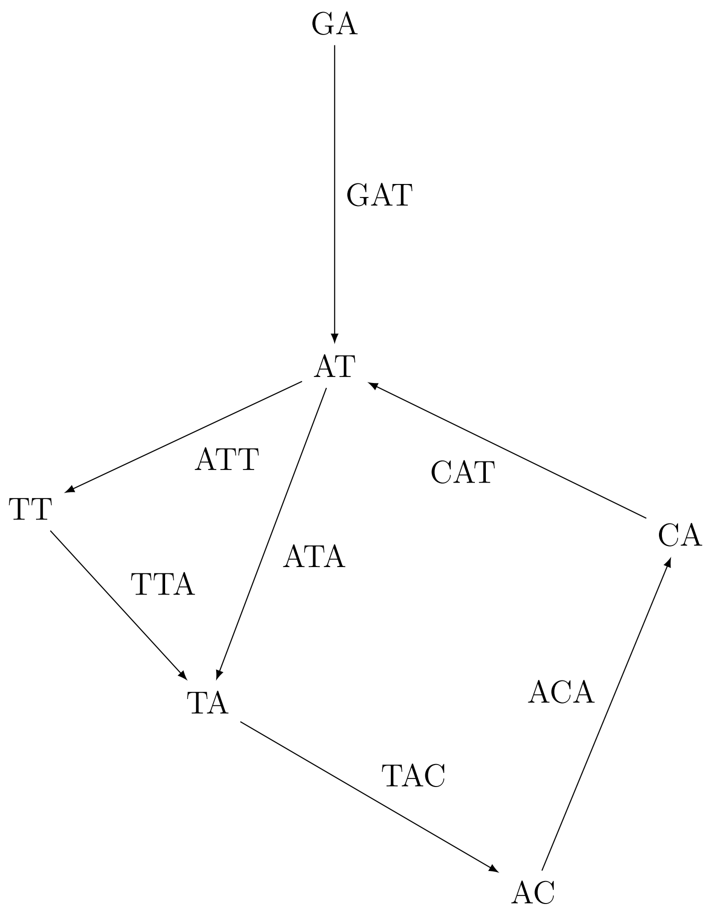
Our first attempt is would be to use these string as a node then connecting an edge to any two node then and each edge as associated cost as the number of overlapping prefix/suffix between the two substring. Then our job would then be to find the path that visit every node and use the maximum number of overlapping character. The path that visit through every node is called hamiltonian path. Finding the maximum cost hamiltonian path is known to be NP hard problem. This translate to a lot of computing power making shot gun sequencing not feasible at all.

However, the breakthrough comes from a CS professor at UCSD, Pavel Pevzner. Instead of using the the substring as vertex. He formulated the k substring as edges and $k-1$ substring as vertex. For example, the sequence

$$\{GAT, ATT, TTA, TAC, ACA, CAT, ATA\}$$

can be turned in to the following (directed) graph.

[6] Prediction: Piti O. 2015.



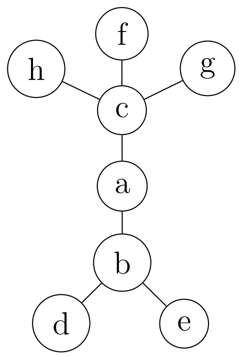
With this graph to find the sequence all we need to do is find an Euler walk on it. What an unexpected place to find Euler walk.[7]

Tree

Def: A tree is a connected graph with no cycle.

Tree has many interesting property. Here is an example of a tree

[7]Here is only the gist of the algorithm there are variants that takes care of repeat/duplication/error and such.



Def: A leaf is a vertex of degree 1.

In the example above d, e, f, g and, h are leaves. Tree has many interesting properties. Let us start with an obvious one.

Theorem: Any connected subgraph of a tree is a tree.

Proof: Pick a connect subgraph of a tree. All we need to show is that is has no cycle.

If there is a cycle in this subgraph then the cycle would exists in the original graph too. But since the original graph doesn't a cycle. The subgraph doesn't have any cycle.

So it is a tree. □

Another interesting property is that the number of edge is fixed by the number of vertices.

Theorem: A tree with n vertices has $n-1$ edges.

Proof: We are going to proof this by induction.

Inductive Predicate: $P(s) =$ All tree of s vertices has $s-1$ edges.

Base Case: Tree of 1 vertex has 0 edge. ✓

Inductive Step: We assume that all graph of i node has $i-1$ for $i < k$.

We want to show that all graph of k node has $k-1$ edges.

Given a graph take away an edge then you get two trees(by the subgraph lemma). Let the two tree has p and q vertices such that

$$p + q = k$$

The tree with $p < k$ vertex has $p - 1$ edges and the tree with $q < k$ vertex has $q - 1$ edges. So the total number of edges is

$$E = p - 1 + q - 1 + \underset{\substack{\uparrow \\ \text{The edge we removed}}}{1}$$

Since $p + q = k$

$$E = k - 1$$

□

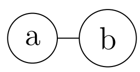
Also you are guaranteed to have at least two leaves.

Theorem: A tree with $n \geq 2$ node has at least two leaves.

Proof: We are going to prove this by induction

Inductive Predicate: $P(s)$ = all graph with s vertices has at least two leaves.

Base Case: Graph with 2 vertex has at least 2 leaves.



Inductive Step: Now we assum that all graphs with i node has at least 2 leaves for $i = 2 \ldots k - 1$.

Now we want to showt that all graphs with k vertex has at least two leaves.

We use the same strategy as we did before. Breaking a tree in to two tree by removing an edge. There are actually two cases base on what happen after we remove an edge.

a.) Case 1. We end up with a tree of size $k - 1$ vertex and a vertex(v). This means that v was a leaf before breaking up.

We know that the $k-1$ vertex tree at least two leaves and connecting v back destroy at most one leave from $k - 1$ vertex and add exactly 1 leaf back to the tree.

So the number of leave is the same as $k - 1$ vertex tree which is greater or equal to 2.

b.) Case 2. We end up with 2 tree of size greater than 1 vertex. Let the two tree be of size p and q both less than k. We know that the tree of size p and q has at least two leaves each. So we have at least four from the two trees. Connecting p and q vertex tree back destroy at most one leaf from each of the tree: we destroy at most two in total. So the original tree has at least 2 leaves.

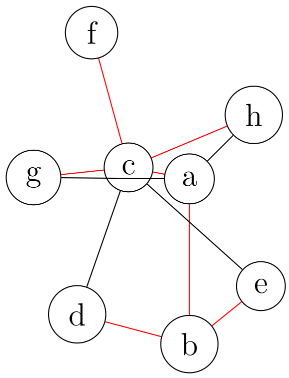
□

Minimum Spanning Tree

Let us consider a type of tree that has numbers of useful application

Def: A spanning tree of a connected graph is a subgraph that is a tree with the same set of vertex as the original graph.

For example the graph with edges shown in red is a spanning tree.



Theorem: Every connected tree has a spanning tree.

Proof: The whole idea of this proof is to say that we can keep removing the cycle while keeping the graph connected.

- Let T be a connected subgraph that has the same number of vertex as G with smallest number of edges.
- Our job is just to show that T is a tree that is T has no cycle and then we are done since T is then our spanning tree.
- If T has a cycle, then we can remove an edge from a cycle and it is still connected. But, that would be T doesn't have the smallest number of edges.
- So, T can't have a cycle. Thus, T is a spanning tree.

□

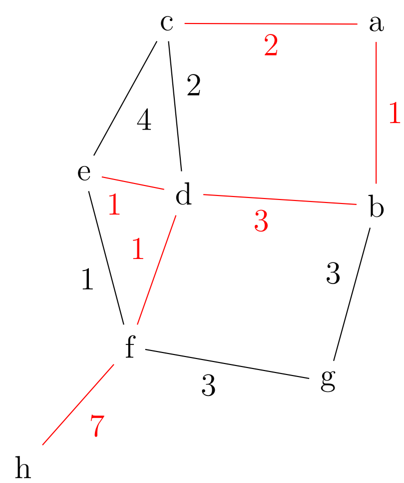
In contrast of most of inductive proofs that do not tell us anything. The proof above actually tells us to find a spanning tree by just keep removing cycle.

But, the most interesting part of spanning tree is what happens when you got a weighted graph.

Def: A edge-weighted graph is a graph with cost associated with edge.

Def: A minimum spanning tree(MST) of a graph is a spanning tree with the minimum sum of the cost on the edges.

The minimum spanning tree can be use to find optimal network route. Optimal distribution network. Here is an example of a minimum spanning tree.



The minimum spanning tree is not unique and there are many ways to find it. All the greedy algorithms work. You can

- Reverse Delete. Keep deleting the edge with highest cost that doesn't disconnect the graph.
- Prim's Algorithm. Start with a node then grow the tree out with the chepest edge that doesn't create a cycle.
- Kruskal's Algorithm. Rank the edge by weight then use the edge with the lowest cost that doesn't create a cycle.

It is easy to show that all these algorithms create a spanning tree. The proof for optimality is a bit involved. Let us consider optimality proof of Kruskal's Algorithm here.

Theorem: The output of Kruskal's Algorithm is a minimum spanning tree.

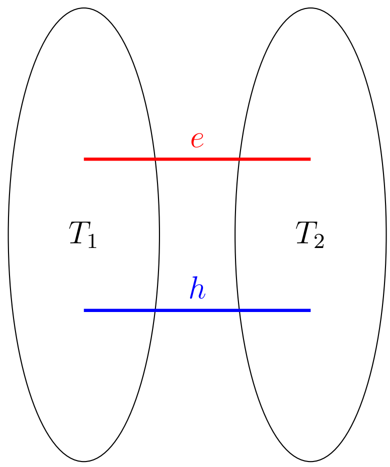
Proof: For simplicity of this proof let us assume that all the weight are different. The goal of this prove is to show that each after we add another edge (e) to the our current graph F. Then, $F + e$ is still a subgraph of some MST. That way we can ensure that each step leads us to optimality.

Base Case: We start with an empty set. Empty set is a subgraph of some MST. ✓

Inductive Step: Assume that F is a subgraph of some MST T. We want to show that if e is a minimum weight edge that doesn't create a cycle at this step then e must be in T. That is $F + e$ is still a subgraph of some MST T.

We are going to do this by contraction. Let us assume e connects two trees T_1 and T_2

- If e is not in T then means that there is an alternative path in T from T_1 to T_2. Let us call this alternative path h.



- But h has to be more costly than e otherwise h is picked already.
- $T - h + e$ connects everything and has lower cost.
- So, e must be in T.

This means every edge we add moves us toward the right answer. □

Connecting Everyone

You can use minimum spanning tree to make sure every home is connected to a telephone grid at the lowest cost. For this problem, you just need to represent vertex as houses, edge as possible connect and the weight would be the cost associated with it. Then, all we need to do is to find the minimum spanning tree. Then, every house will be connected at the lowest cost.

Clustering

One application of minimum spanning tree is in clustering. We can use each vertex to represent the data point and the edge to represent the similarity between pair of data point.

Fiding an minimum spanning tree of this graph will allow us to break data point into clusters.

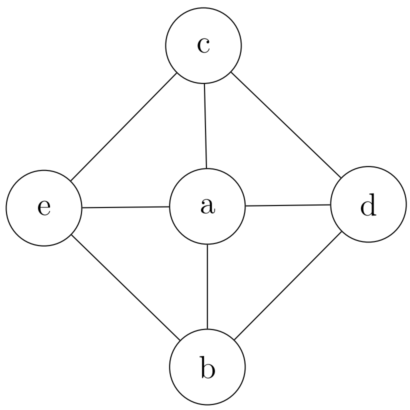
You can also use this in image segmentation. The idea is similar to clustering the vertices are pixels. The pixel are connected only if they are adjacent. The weight would be the difference in color of each pixel. With minimum spanning tree you can then cluster image by color thereby segmenting the image.

Planar Graph

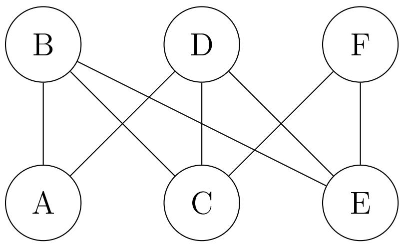
The formal definition of planar graph is a bit involved. Let us use a less formal definition here.

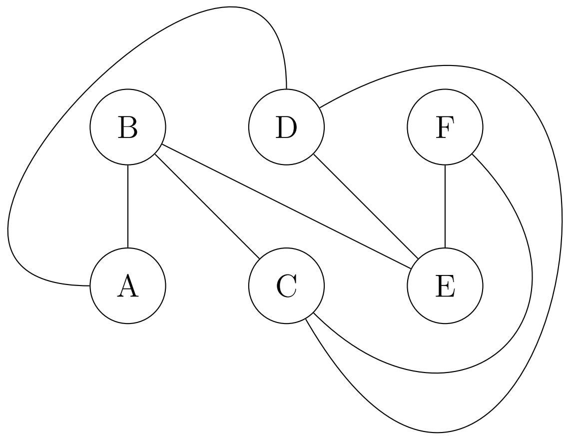
Def: A planar graph is a graph that you can draw on a paper such that no two edges intersect.

For example, the following graph is planar.

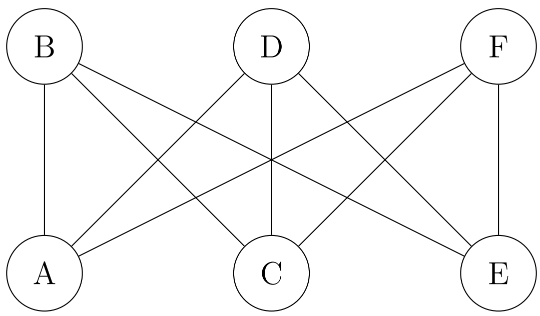


Sometimes it is not trivial to see which graph is planar. For example, the following graph is also planar.



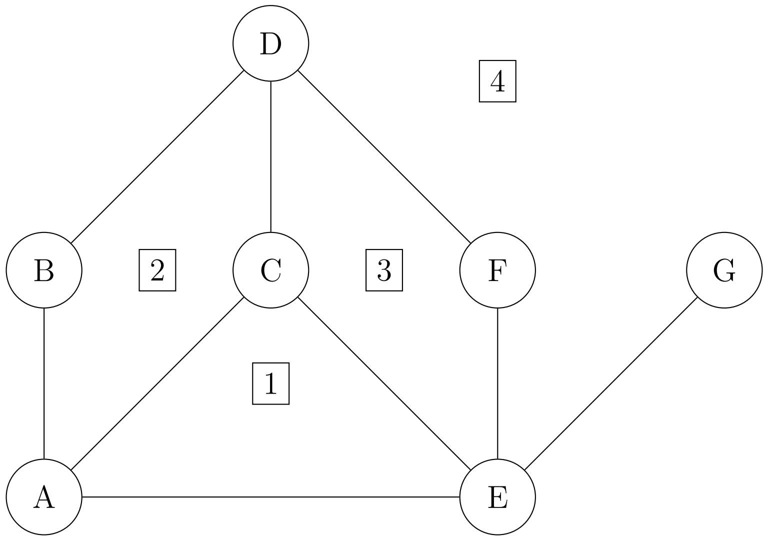


But the following graph is not a planar graph.



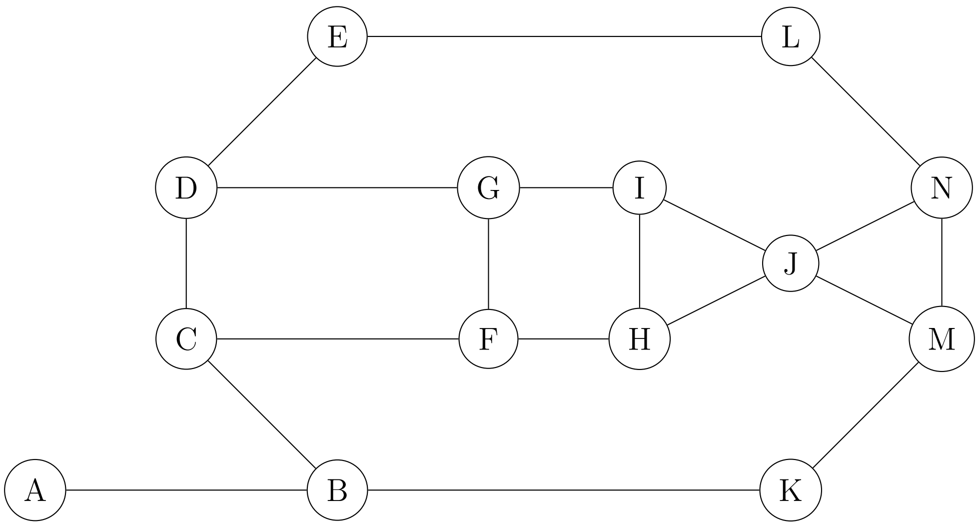
Euler Formula

Planar graph divide the area in to segments. We call these segments faces. The name will become clear later on. For example the following graph has 4 faces. We count the area outside also.

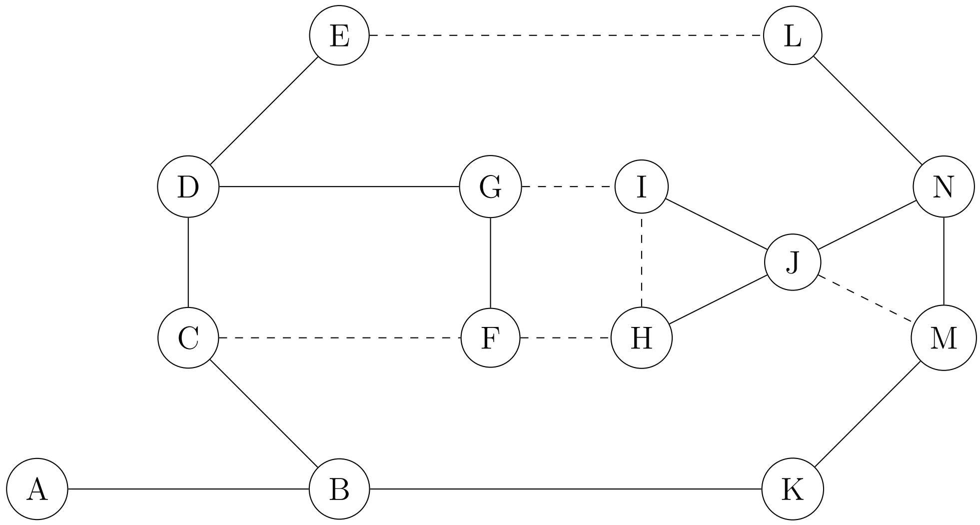


After drawing a few graph you will notice that there seems to be a relation among the number of faces(f), number of edges(e) and the number of vertices(v).

Let us consider a bed time story. Suppose you have a fortress you connect each watch tower with a wall. All the region outside is water and your goal is to break the wall such that all regions inside are flooded.



- The number of basin including ocean is $f = 7$.
- To get every region flodded you need to break $f - 1$ walls. This is because each wall you break you flood one more region. This means you break all the cycle.



- So the graph that is left over is a tree. We know that the number of edge for a tree is

$$e_{\text{left over}} = v - 1$$

- We know that the total number of walls(e) is the sum of the number of wall we destroy($f-1$) and the number of walls we have left($v-1$).

$$e = (f-1) + (v-1),$$

which simplifies to a relation called Euler's Formula.[8] If a graph is planar then

$$f + v = e + 2$$

Maximum Number of Edges

You may notice that the the reason that a graph is not a planar graph is because it just have too many edges for the given number of vertex. Let us find the maximum number of edges a v vertex planar graph can have.

- We know that each face need at least 3 edges and each edge contribute to 2 faces.

$$f \le \overset{\substack{\text{each edge is used by 2 faces}\\ \downarrow}}{2}\frac{e}{\underset{\substack{\uparrow\\ \text{each face use 3 edges}}}{3}}$$

- From Euler's Formula we have

$$f = e - v + 2$$

[8] There are actually two Euler's formulas. The other one is the relation between exponential of complex number to trigonometric functions.

- This means

$$\begin{aligned} e - v + 2 \leq& \frac{2e}{3} \\ \frac{1}{3}e \leq& v - 2 \\ e \leq& 3v - 6 \end{aligned}$$

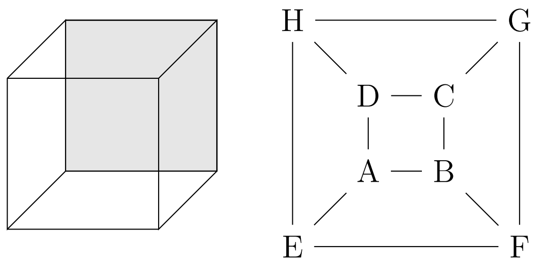
- So the maximum number of edge a planar graph can have is

$$e \leq 3v - 6$$

Why outside and Why faces?

The name faces and the fact that we include the outside region in faces seem like a non intuitive way to define faces. The name actually comes from geometry. The problem of counting faces of polygon.

Let us consider a cube.



It has 8 vertices 12 edges and 6 faces.

$$f + v = 6 + 8 = 12 + 2 = e + 2$$

This is actually not a coincidence. The reason is that you can turn any polygon into planar graph by poking a hole on one side and strech it out. For the graph shown above we poke a hole on the gray side and flatten it out. The gray face then become the outside area. All the other faces become the internal regions. Thus, the number of region including the outside is the number of faces.

Try it yourself with a tetrahedron or a square pyramid.

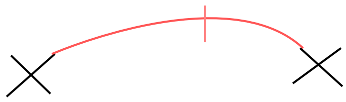
Brussel Sprouts Game

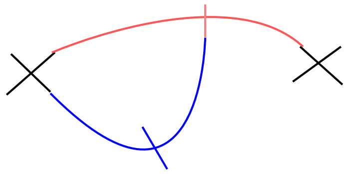
Let us consider a game of two player called Brussel Sprouts.

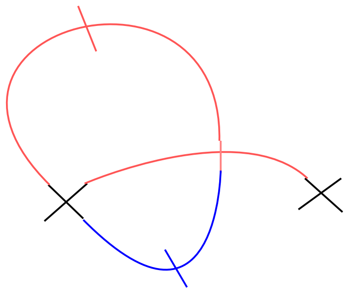
- We start the game with 2 crosses.

- Each player alternate drawing a line between two free ends with out crossing the previously drawn line. Each time each player draw a line the player then make a cross in the middle creating more free ends.

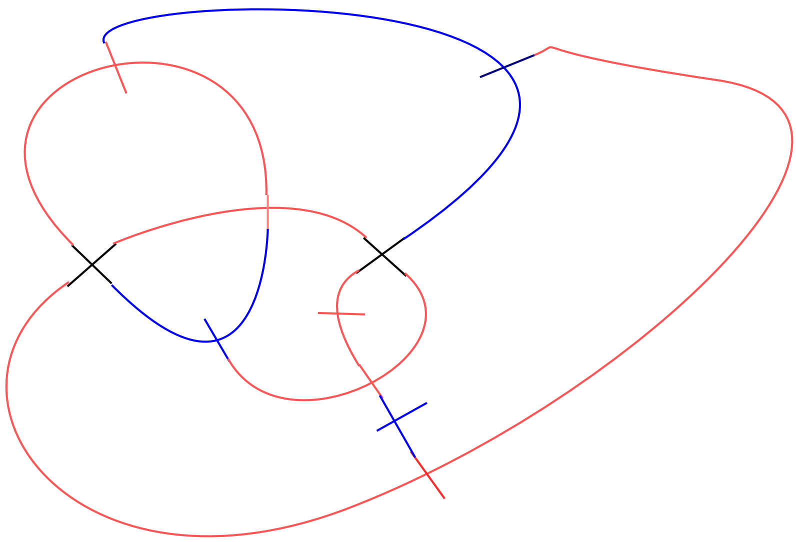
  Here are some example of the first few turns of the gameplay







- The game ends when there is no more possible move. The player that draw the last valid line wins. Here is an example of the board at the end.



After playing this a few time, you will probably realize that not only that the second player always win. The game always end in exactly 8 turns. Let us this.

Theorem: The number of free end is always 8.

Proof: First, the number of free ends never change. Each time we draw a line we destroy to free end one for each end of the line. Then, we create two more by crossing the middle of a line.

Since we start with 8 free ends, we will always have 8 free ends. □

Theorem: The game ends after exactly 8 turns.

Proof: Let assume that we are at turn m. Let us start by counting the number of vertex and edges.

- Since we start with 2 vertex and each time we draw a line we create one more vertex. Thus,
$$v_m = 2 + m.$$
- For the number of edges, since we start with 0 edge. At each turn we create two more edges (one left and one right of the newly created cross). Thus,
$$e_m = 2m.$$
- Now we need the condition for the end of the game. The game ends when each face has exactly one free end in it. If it has more than one free end, then you can continue the game. So the condition for end of game is
$$\#\text{ faces} = \#\text{ free end}$$
- We know from the previous theorem that the number of free end is always 8. This means that the game ends when we have 8 faces.
- Now our job is to find the number of turn needed to create 8 faces. We can use euler formula for this
$$f + v = e + 2$$
That means at the end of the game where $f = 8$ the turn number m can be found from
$$8 + (2 + m) = 2m + 2$$
Solving for m gives
$$m = 8$$

- This means that the 8th turn is the last turn of the game. So the second player always win.

□

Matching Problem

Stable Marriage

Let us consider a situation where there are two married couple: (Alice, Bob) and (Charles, Dorothy). Let us suppose that Alice like Charles more than her husband Bob. Moreover, Charles like Alice more than his wife Dorothy. If this situation happens Alice and Charles are going to ditch their spouse and run off together. Such marriages are called unstable and Alice and Charles is called a rogue couple.

Let us consider a village far away where there are equal number of boys and girl. You are a little cupid. Your job is to match boys and girls together such that there is no rogue couple. So, people in the village can live happily ever after.

Mating Ritual

After a couple hour of deep thought. You came up with a mating ritual which goes like the following

a.) First you have every boys ranks the girls according to his preference. The girls also make their list of boys according to her preference.

b.) On the first day, each girl wait at her balcony. Each boy then walk to his favorite girl on the list.

  - Each girl then looks at all the boys at her balcony and pick her favorite.
  - Each girl tells all the boys at her balcony that she did not pick to never come back again. The boy then cross the girl's name off his list.
  - Each girl tells the boy she pick for that day to "come back tomorrow and that she may marry him."

c.) On the subsequent days the process is then repeated. Each boy go to his favorite girl that hasn't been crossed out yet. The girl then pick her favorite tell him to wait and tell everyone else to go away.

d.) The ritual ends when there is exactly one boy at each balcony.

Ritual Eventually Ends

First thing that probably come to our mind is that is this ritual a troll? We may get into an infinite loop. Actually, we are guarantee that the ritual will end after finite steps.

The technique we use here is a general technique to show that some algorithm actually terminates. The idea is to monitor some finite resource and to show that the finite resource we have is strictly decreasing(at steady rate). Then we will run out of resource in finite number of step.

The resource we are considering here is the number of girls on every boys list. If the ritual doesn't end yet, that means there is at one girl that has 2 or more boys at her balcony. This means one boy has to cross out a girl from his list. Since the number of girls on boys list is finite. The ritual will eventually end.

You also know that everyone will get married at the end by pigeon hole. If there is empty balcony then there is at least one balcony with 2 boys which means the ritual doesn't end yet.

Every Marriage is Stable

Another question you may ask is whether the pairing provide by this ritual is actually stable. We will prove this by showing that for any pair or boy and girl who are not married at the end of the ritual are not rogue couple.

Let us consider Alice and Bob who are not married at the end of the ritual. There are two cases.

- Case 1. Alice is not on Bob's list at the end. This means that Bob crossed Alice out on the day Alice found a better guy(Charles). Since we know that Alice must prefer her husband at least as much as Charles. We know that Alice must prefer her husband over Bob. So, Alice will never run away from her husband.
- Case 2. Alice is still on Bob's list at the end. This means that Bob is stuck with a girl whom he prefer over Alice. That means Bob prefer his wife over Alice. So, Bob will not run away from his wife.

□

Life is Good for Boys

This ritual seem like a really good deal for girls. Afterall, the girls get to pick her favorite boys that comes to her balcony while the boys keeps getting rejected go home crying over and over again.

To get and idea of who is getting a better deal. Let us consider

Def: Realm of possible Spouses of a person A. Let us consider all possible stable matching on the preference list. Then we list all spouses of A on all those stable matching. Then the set is called realm of possible Spouses.

Person A can also rank all spouses in his/her realm of possible Spouses. The one with the highest rank is the best possible spouse and the one with the lowest rank is worst possible spouse.

Theorem: The mating ritual marries every boy to his best possible spouse.

Proof: We will prove this by contradiction. Let us assume for the sake of contradictino that some boy doesn't not get married to his best possible spouse Alice.

- This means that there is the first day where a boy cross out his best possible spouse. Let us call the boy (Bob) and the girl (Alice).
- Since Bob crosses out Alice that mean on that day Alice met someone better than Bob. Let us called that person Charles. So we know that

  Alice like Charles more than Bob.

- Since this is the first day that anyone crosses out their best possible spouse. Charles has not cross out his best possible spouse (Dorothy) or in other words Charles has not visit his possible spouse. This means

  Charles likes Alice more than Dorothy.

- Now since Alice is Bob's best possible spouse, there is a stable marriage which Alice is married to Bob.

  However, in this stable marriage Charles will get matched with someone which he ranks at best as high as Dorothy since Dorothy is Charles' best possible spouse. This means Charles still likes Alice more than his wife.

- Since Alice like Charles more than her husband and Charles like Alice more than his wife. Charles and Alice are rogue couple in this stable marriage. Thus we have a contradiction.
- Therefore, in mating ritual, every boy gets married to his best possible spouse.

□

Not so much for Girl

Theorem: The mating ritual marries every girl to her worst possible spouse.

Proof: Let us show this by contradiction.

- Let us supposed that the mating ritual marries a girl (Alice) to someone (Bob) whom she likes more than her worst possible spouse(Charles). So we know that

  Alice likes Bob more than Charles.

- From the previous theorem we know that Alice must be Bob's best possible spouse since the mating ritual match them together. This means

  Bob prefer Alice to everyone in realm of possible spouse.

- Let us consider a stable marriage M[9] where Alice is married to Charles. We know that such stable marriage exists since Charles is worst possible spouse for Alice. Thus

  In M, Alice is married to Charles whom she likes less than Bob.

- In M, Bob has to be married to someone whom he like less than Alice since Alice is his possible spouse and Alice is already married to someone else.

  In M, Bob is married to someone whom he like less than Alice.

- That means that in stable marriage M, Bob and Alice are rogue couple. Therefore, we have our contradiction.
- Thus, mating ritual marries every girl to her worst possible spouse. Poor girls.

□

[9]Not produced by the mating ritual.