Combinatorics: The Fine Art of Counting

Week 9 Lecture Notes – Graph Theory

For completeness I have included the definitions from last week's lecture which we will be using in today's lecture along with statements of the theorems we proved.

Definitions

Graph: A *graph* G = (V, E) consists of an arbitrary set of objects V called **vertices** and a set E which contains unordered pairs of distinct elements of V called **edges**.

Adjacent: Two vertices in a graph are *adjacent* if there is an edge containing both of them. Two edges are adjacent if they contain a common vertex. Adjacent vertices are called neighbors.

Degree: For any vertex v in a graph, the *degree* of the vertex is equal to the number of edges which contain the vertex. The degree of v is denoted by *d(v)*.

Regular Graph: A graph in which every vertex has the same degree is called a *regular* graph. If all vertices have degree k, the graph is said to be **k-regular**.

Complete Graph: The *complete graph* on n vertices $\mathbf{K_n}$ consists of the vertex set V = $\{v_1,v_2,\ldots,v_n\}$ and the edge set E containing all pairs (v_i,v_j) of vertices in V.

Isomorphic: Two graphs are *isomorphic* if there exists a one-to-one correspondence between their vertex sets (i.e. a re-labeling) which induces a one-to-one correspondence between their edge sets. More formally, if L is a re-labeling which maps the vertices of G to the vertices of H, then the edge set of H is precisely the set of edges (L(v),L(w)) where (v,w) is an edge in G.

Sub-graph: A graph $G_1 = (V_1, E_1)$ is a *sub-graph* of $G_2 = (V_2, E_2)$ whenever $V_1 \subseteq V_2$ and $E_1 \subseteq E_2$.

Path: A *path* of length n is the graph $\mathbf{P_n}$ on *n+1* vertices $\{v_0, v_1, v_2, \ldots, v_n\}$ with *n* edges (v_0,v_1), (v_1,v_2), …, (v_{n-1},v_n).

Cycle: A *cycle* of length n is the graph $\mathbf{C_n}$ on *n* vertices $\{v_0, v_2, \ldots, v_{n-1}\}$ with *n* edges (v_0,v_1), (v_1,v_2), …, (v_{n-1},v_0).

We say that a given graph **contains** a path (or cycle) of length n if it contains a sub-graph which is isomorphic to P_n (or C_n).

Connected: A graph that contains a path between every pair of vertices is *connected*. Every graph consists of one or more disjoint connected sub-graphs called the **connected components**.

Distance: The *distance* between two connected vertices is the length of the shortest path between the vertices.

Diameter: The *diameter* of a connected graph is the maximum distance between any two vertices in the graph.

Forests and Trees: A graph which does not contain a cycle is called a *forest*. If it is a connected graph, it is called a *tree*. The connected components of a forest are trees.

End-points and Isolated Vertices: An *end-point* is a vertex with degree 1. An *isolated* vertex is a vertex with degree 0.

Hamiltonian Graph: A graph which contains a **Hamiltonian cycle**, i.e. a cycle which includes all the vertices, is said to be **Hamiltonian**.

Walks, Trails, and Circuits: A *walk* in a graph is a sequence of adjacent edges. A *trail* is a walk with distinct edges. A *circuit* is a trail in which the first and last edge are adjacent.

Eulerian Graph: A trail which includes all of the edges of a graph and visits every vertex is called an **Eulerian Tour**. If a graph contains an Eulerian tour which is a circuit, i.e. an **Eulerian circuit**, the graph is simply said to be **Eulerian**.

Theorems Proven Last Week

Theorem 1: The sum of the degrees of all the vertices in a graph is equal to twice the number of edges, i.e. $\Sigma d(v) = 2|E|$

Theorem 2: Every tree with at least one edge contains two end-points

Theorem 3: A graph with n vertices is a tree if and only if it is connected and has n-1 edges.

Theorem 4: A graph is Eulerian if and only if it is a connected graph in which every vertex has even degree.

Corollary 4.1: A graph contains an Eulerian tour if and only if it is a connected graph with at most two vertices of odd degree.

New Material

We begin with a simple corollary to theorem 3 which follows almost immediately.

Corollary 3.1: A graph with n vertices and at least n edges contains a cycle.

Proof: Let G be a graph with n vertices. If G is connected then by theorem 3 it is not a tree, so it contains a cycle. If G is not connected, one of its connected components has at least as many edges as vertices so this component is not a tree and must contain a cycle, hence G contains a cycle.
QED

This simple fact has a lot of practical applications, e.g. the sleepy mathematicians USAMO problem mentioned previously. In many situations it may be difficult to find a particular cycle, but just by counting edges we can prove that a cycle must exist.

There are a few more basic facts about trees that are useful to know.

Theorem 5: A graph is a tree if and only if there is a unique path between any two vertices.

Proof: We have two things to prove, the "if" and the "only if". Suppose G is a tree containing vertices v and w. G is connected so there is a path from v and w, we simply need to show that this path must be unique. Suppose there were two distinct paths from v and w. Starting from v, let s be the first vertex where these paths diverge. s could be equal to v, but it cannot be equal to w since the paths must diverge somewhere. Let t be the first vertex where the paths meet again – there must be such a vertex since they meet at w. The two segments of the paths between s and t are disjoint, so together they form a cycle containing s and t which contradicts our supposition that G is a tree. Thus the path from v to w must be unique.

Conversely, suppose G is a graph which contains a unique path between any two vertices. G is clearly connected. G cannot contain a cycle, because a cycle contains two distinct paths between any pair of vertices in it. Therefore G is a tree.
QED

Trees are very nice graphs to work with. Unfortunately, not all graphs are trees. However all connected graphs contain a tree which includes all the vertices. Such a graph is called a spanning tree.

Spanning Trees: A *spanning sub-graph* of G is a sub-graph H which includes all the vertices of G. In the case where H is a tree, H is called a *spanning tree*.

Theorem 6: Every connected graph contains a spanning tree.

Proof: Let G be a connected graph. If G is a tree we are done, otherwise G must contain a cycle. Removing an edge from this cycle will result in a connected graph with the same vertex set as G but fewer edges. We can continue in this manner until there are no more cycles (there are only a finite number of edges to remove), at which point the remaining graph must be a tree.
QED

This proof gives us one way to construct a spanning tree, namely by removing edges from cycles. This is not a particular good method if the graph contains a lot of edges. A more efficient approach is to simply pick a vertex to be a seedling, and then "grow" the spanning tree by connecting vertices one at a time. As long as not all the vertices are in the tree, there must be a vertex in the tree which has a neighboring vertex not in the tree – simply pick one such vertex, connect it to the tree with a single edge, and repeat until all the vertices are in the tree.

Spanning trees are useful in a lot of situations – efficiently broadcasting a message to all nodes in a network is but one example.

We now look at perhaps the simplest case of graphs which aren't trees, graphs where every vertex has degree 2.

Theorem 7: A graph is 2-regular if and only if all its connected components are cycles.

Proof: One direction of the theorem is trivial – a graph whose connected components are all cycles is clearly 2-regular. We prove the other direction by induction on the number of vertices in the graph. The base case is K_3 which is 2-regular and has one connected component which is a cycle. For a graph with n > 3 vertices, note that by Theorem 1, a 2-regular graph with n vertices has n edges and by Corollary 3.1 such a graph must contain a cycle. Since the graph is 2-regular, none of the vertices in this cycle can be contained in any other edges, so the cycle is a connected component. The remainder of the graph (if any) is 2-regular and has less than n vertices so the inductive hypothesis applies.
QED

The theorem above is a very special instance of a much more general result regarding graphs which can be decomposed into cycles, but first we need to define exactly what we mean by this.

Decomposition: A *decomposition* of a graph G is a partitioning of the edges of the graph among a collection of sub-graphs H_1, H_2, H_3, …, H_n.

Note that the partitioning applies only to the edges, not the vertices, so the sub-graphs H_i can share vertices, just not edges. Any isolated vertices are effectively ignored by a decomposition, since it only pertains to the edges (according to our definition, isolated vertices could be included in any, all, or none of the sub-graphs in the decomposition). We say that a graph G has a **decomposition into cycles** if there exists a decomposition $H_1, \ldots, H_n$ of G where every H_i is isomorphic to a cycle.

Theorem 8: A graph has a decomposition into cycles if and only if every vertex has even degree.

Proof: We effectively proved the "if" part of the statement in Theorem 4 when we showed that a graph in which every vertex has even degree can have a cycle removed from it leaving a graph with fewer edges that also has all vertices with even degree. The cycle we removed, will be one of the sub-graphs in the decomposition, and we can simply keep removing cycles until we have partitioned all the edges.

Proving the "only if" part simply requires noting that if G is a graph with a decomposition into sub-graphs, the degree of any vertex in G is simply the sum of its degree in each of the sub-graphs in the decomposition which contain it, since they don't share any edges. If the sub-graphs are all isomorphic to cycles, the degree of any vertex in any sub-graph is 2, and the sum of the degrees is even.
QED

Corollary 8.1: A graph is Eulerian if and only if it is connected and has a decomposition into cycles.

This follows immediately from Theorem 4 and 8. Being able to decompose a graph into cycles is extremely useful in many applications and is one of the main application of Eulerian graphs. The results above can be generalized to handle decompositions of graphs with vertices of odd degree into cycles and paths, but we will leave this topic for now.

Graph Coloring

Graph coloring is a major sub-topic of graph theory with many useful applications as well as many unsolved problems. There are two types of graph colorings we will consider.

Vertex-Colorings and Edge-Colorings: Given a set C called the set of colors (these could be numbers, letters, names, whatever), a function which assigns a value in C to each vertex of a graph is called a *vertex-coloring*. A proper vertex-coloring never assigns adjacent vertices the same color. Similarly, a function which assigns a value from a set of colors C to each edge in a graph is called an *edge-coloring*. A proper edge-coloring never assigns adjacent edges the same color.

In the case of vertex-colorings, we will primarily be interested in colorings which are proper, and following convention, we will use the word **coloring** to mean a proper vertex-coloring. In contrast, we will want to consider edge-colorings which are not necessarily proper.

Note that the values of the set C are arbitrary, what is important is the size of C. The most interesting question we will consider regarding colorings is how big the set C must be in order for a coloring of a given graph to exist.

k-Coloring: A coloring of a graph using a set of k colors is called a *k-coloring*. A graph which has a k-coloring is said to be **k-colorable**.

The four-color theorem is equivalent to the statement that all planar graphs are 4-colorable. Note that a graph which is k-colorable might be colorable with fewer than k colors. It is often desirable to minimize the number of colors, i.e. find the smallest k.

Chromatic Number: The chromatic number of a graph G is the least k for which a k-coloring of G exists.

Thus if a graph G has chromatic number k, then G has a k-coloring, but not a (k-1)-coloring. For example a path has chromatic number 2, while the complete graph K_n has chromatic number n. We now consider the chromatic number of cycles.

Theorem 9: The chromatic number of C_n is 2 if n is even, and 3 if n is odd.

Proof: First note that the chromatic number must be at least 2 for any graph which has an edge in it, including all cycles. We now prove the theorem by induction on n. We will consider two base cases, C_3 and C_4. C_3 is isomorphic to K_3 which has chromatic number 3. C_4 can be colored with two colors by giving opposing corners of the square the same color. For n > 4, we can take a coloring of C_{n-2} and insert 2 adjacent vertices and edges and then color the new vertices appropriately to get a coloring of C_n. Thus the chromatic number of C_n is not greater than that of C_{n-2}. In the case where n is odd, note that if C_n had chromatic number 2, we could remove two adjacent vertices and edges to get a 2-coloring of C_{n-2} which contradicts the inductive hypothesis since n-2 must be odd if n is odd.
QED

Graphs which are 2-colorable are sufficiently important that they have a special name.

Bipartite Graph: A graph which is 2-colorable is called *bipartite*.

We have already seen several bipartite graphs, including paths, cycles with even length, and the graph of the cube (but not any other regular polyhedra)

Complete Bipartite Graph: The complete bipartite graph $K_{m,n}$ consists of a vertex set which may be partitioned into two subsets of size m and n, along with all edges which contain exactly one vertex from each subset. Note that the graph $K_{m,n}$ has m+n vertices and m*n edges.

Graphs which are bipartite have a surprisingly simple characterization.

Theorem 10: A graph is bipartite if and only it does not contain a cycle of odd length.

Proof: Note that by Theorem 9, a cycle of odd length has chromatic number 3. The chromatic number of any graph must be at least as big as the chromatic number of any of its sub-graphs, so a graph containing an odd cycle can't be bipartite. The other direction of the theorem is more interesting.

Suppose G is a graph which does not contain any odd cycles. Without loss of generality we can assume that G is connected, since to show that a graph is bipartite it is enough to show that each of its connected components is bipartite.

Pick a vertex v and color it red. For every other vertex x, color x red if d(x,v) is even, otherwise color it blue. We now need to show that this is a proper coloring. Suppose not, i.e. suppose (x,y) is an edge in G and x and y both have the same color. Note that d(x,v) and d(y,v) are either equal or differ by one, since x and y are adjacent. We will show that these distances must be different, which implies that x and y can't have the same color.

Consider a shortest path from v to x and a shortest path from v to y (one of these paths could have length 0 if x or y is equal to v – this doesn't affect the proof). Let w be the last vertex the

paths have in common. Note that the length of both paths from v up to w must be d(v,w) since otherwise one of them would not be a shortest path. If w is equal to x or y, then clearly one of the paths is longer than the other. Otherwise, consider the cycle formed by taking the path from w to x together with the path from w to y and the edge (x,y). The length of this cycle is d(v,x)-d(v,w) + d(v,y)-d(v,w) + 1. This must be an even number since G does not contain an odd cycle, which implies that d(v,x) and d(v,y) cannot be equal.
QED

Note that a special case of this theorem is that all trees are bipartite, since they don't contain any cycles (odd or otherwise).

The proof above not only characterizes the graphs which are 2-colorable, but gives an efficient method for checking whether this is true. First pick a vertex v and compute the distance from v to every other vertex in the graph using the algorithm below. Once we have done this we then color the vertices with even distances red (including v itself) and color the other vertices blue. We then simply check whether this coloring is proper by examining the end-points of each edge and making sure they have different colors. If we find two adjacent vertices with the same color than by the proof above, we know that the graph is not 2-colorable.

The algorithm below gives a simple method for computing the shortest path from v to any vertex in the graph G. There are other algorithms, but this one has the virtue of simplicity and most others are variations of the same basic idea.

Shortest Path Algorithm

Given a graph G and a vertex v, we wish to compute the value d(x) = d(v,x) for all x in G. We will do this by starting with a worst case estimate of d(x) for each vertex x, call it e(x), which is always greater than or equal to d(x), and then proceed to refine our estimate step by step until e(x) = d(x) for every vertex x. Let n be the number of vertices in G.

> **Initialization:** Set e(v) = 0 and set e(x) = n.
> **Update Step:** For each vertex x, check e(y) for each adjacent vertex y, and if e(y)+1 is less than e(x), set e(x) = e(y)+1 and note that a change was made.
> **Termination:** If no changes were made for any vertex, terminate.

It is easy to see that e(x) ≥ d(x) always holds, since this is true initially and the update step will never make it false – if a neighboring vertex has a valid distance estimate more than one less than our current estimate, we could use a path that included the neighboring edge followed by a path from the neighbor to v. It is also clear that the algorithm must eventually terminate, since if a change is made than e(x) is reduced by at least 1 for some vertex x. The total sum of the e(x)'s is thus always decreasing and can never be 0. In fact the sum of all the e(x)'s is less than n^2 at the beginning, so the algorithm must terminate after fewer than n^2 update steps.

The fact that may not be immediately obvious is that when the algorithm terminates, e(x) = d(x) for every vertex x. To see why this must be true, suppose it is not. Then e(x) > d(x) for some vertex x. Now walk along the shortest path P from v to x and find the first edge (y,z) where e(y) = d(y) but e(z) > d(z) – such an edge must exist since e(v) = d(v) but e(x) > d(x). Any sub-path of a shortest path is also a shortest path, so in particular the segment of P from v to y is a shortest path from v to y a and the segment of P from v to z is a shortest path from v to z, hence d(z) > d(y). But e(z) > d(z) > d(y) = e(y), so e(y)+1 must be less than e(z). This means that the algorithm cannot have terminated, because it would have changed e(z) in the update step.

This algorithm can be used to compute the shortest path between any two vertices reasonably efficiently, so finding the optimal route between two points on a map is not a particularly hard problem. This problem should not be confused with the traveling salesman problem which aims to find the optimal route which travels through an entire set of points on a map. As the set of points to be visited gets large, the problem becomes very difficult.

Surprisingly, in contrast to determining whether a graph is bipartite, the problem of determining whether a graph is 3-colorable is much harder. It belongs in the same class of NP-complete problems as the traveling salesman and Hamiltonian path problems. No good solution is known and trying all possible colorings is impractical for all but the smallest graphs.

Edge Colorings

Suppose you are organizing a tournament for n contestants using a round-robin format, i.e. each player will be matched against every other player at some point during the tournament which will consist of several rounds. How many rounds will be required to accomplish this? How should the contestants be matched up in each round? What if the number of contestants is odd, how should this be handled? The answer to this and many other questions can be found using edge-colorings of graphs.

Given a graph G with its edges colored, for any particular color c, we can define a sub-graph H_c consisting of just those edges with the color c and the vertices they contain. A sub-graph in which all the edges have the same color is called **mono-chromatic.**

Any coloring of the edges partitions the edges and thus gives a decomposition of G into mono-chromatic sub-graphs. In the case of a proper edge-coloring, these mono-chromatic sub-graphs must consist entirely of non-adjacent edges and their end-points, i.e. the sub-graphs match-up pairs of vertices. Every vertex in such a sub-graph will have degree 1.

Matching: Given a graph G, a 1-regular sub-graph H of G is called a *matching*. If H spans G (i.e. has the same vertex set), then H is called a **perfect matching**.

Such a decomposition is exactly what we need to schedule a tournament with n players. If we have a proper edge-coloring of the graph K_n with k colors. We can use the coloring to schedule a tournament with k rounds – the mono-chromatic sub-graphs will tell us how to match the players in each round.

Note that each vertex in K_n has n-1 edges, so k must be at least n-1. This is true for any regular graph, i.e. a proper edge-coloring of a k-regular graph must use at least k colors. When a k-regular graph can be colored with exactly k colors, a remarkable thing happens – every vertex appears in every mono-chromatic sub-graph, i.e. the coloring decomposes the graph into k perfect matchings.

As an example, consider the graph K_4. The edges of K_4 can be properly colored with 3 colors. This is easy to see if we draw K_4 as an equilateral triangle with a vertex in the center connected to each of the corners oriented so that it's base is horizontal Take the horizontal along with the vertical edge at the top and color them R, then take the congruent pair of edges which are rotated 120 degrees and color them G, and then color the final pair (rotated another 120 degrees) blue. The cube is another example of a 3-regular graph whose edges can be properly colored with 3 colors.

The example of K_4 described above can be generalized to give the following remarkable result:

Theorem 11: K_n has a proper edge-coloring with n-1 colors if and only if n is even.

Proof: It is easy to see why n-1 colors is not sufficient when n is odd. K_n is a regular graph of degree n-1. As noted above, if it is properly colored with n-1 colors, then the mono-chromatic sub-graphs will be perfect matchings, but it is not possible to perfectly match an odd number of vertices. Now suppose that n is even. We will construct a perfect matching using a very useful trick of the graph theory trade called **"turning"**.

Pick one vertex x of K_n and then place the other m=n-1 vertices in an evenly spaced circle around x with vertex 0 directly above x and the remaining vertices numbered from 0 to m-1. Draw a vertical edge (0,x), then draw horizontal edges (1,m-1), (2,m-2), …, ((m-1)/2,(m+1)/2) to pair up all the vertices. This set of edges forms a perfect matching and will be our first mono-chromatic sub-graph. Now rotate all the edges clockwise around the circle by one vertex to get the next mono-chromatic sub-graph and continue all the way around the circle. Because m is odd, no edges will be repeated and every edge will be colored exactly once using m=n-1 colors.
QED

The "turning" trick used in the proof above can be used in lots of other situations. For example the octahedron can be properly edge-colored using 4 colors by standing the octahedron on a vertex with a horizontal edge facing you. Take the right-hand edge leading down from the top vertex along with the left-hand edge leading up from the bottom vertex together with the horizontal edge in back as the first perfect matching, then rotate this set of edges 90 degrees to get the next perfect matching.

The last topic we will look at relates to edge-colorings which are not proper colorings.

Ramsey Theory

One of the simplest and most well known results in graph theory (sometimes known as the "strangers and friends" theorem) can be stated informally as follows:

"At a party with six people, either there are three mutual friends or three mutual strangers (or both)."

We can state this in graph-theoretic terms as follows:

Theorem 12: Any 2-coloring of the edges of K_6 contains a mono-chromatic triangle.

Proof: Given a 2-coloring of the edges of K_6 pick any vertex v. Since v has degree 5 at least 3 of the edges containing v must be the same color, call it red. (this is a trivial example of what is sometimes known as the "**pigeon-hole-principle"** – if 5 pigeons are crammed into two boxes, one of the boxes must have at least 3 pigeons in it). Now consider 3 neighbors of v which are joined to v via a red edge. Either the 3 edges between the 3 neighbors are all blue, in which case there is a blue triangle, or one of them is red, in which case that edge together with the edges leading to v constitute a red triangle.
QED

Given the proof, the theorem above may seem weak – the argument we made applied to every vertex and we only needed to use it once. We can in fact prove something stronger, that there must be at least two mono-chromatic triangles. It is possible to prove this using a much more involved version of the argument above, but instead we will give a very different proof based on counting.

Theorem 12.1: Any 2-coloring of the edges of K_6 contains two mono-chromatic triangles.

Proof: Given a 2-coloring of the edges of K_6 call a triangle bi-chromatic if it contains edges in both colors. We know that K_6 contains (6 3) = 20 triangles, but how many of these are bi-chromatic? We will show that at most 18 are, so there must be 2 mono-chromatic triangles.

Each bi-chromatic triangle contains exactly two bi-chromatic paths of length 2, and every bi-chromatic path of length 2 is contained in exactly 1 bi-chromatic triangle. Thus the number of bi-chromatic 2-paths is exactly twice the number of bi-chromatic triangles. A given vertex can be the center of at most 6 bi-chromatic paths (e.g. when 3 edges are red and 2 are blue). There are 6 vertices in K_6 so there are at most 6*6 = 36 bi-chromatic paths which means there at most 18 bi-chromatic triangles.
QED

Note that the bi-chromatic triangles in the theorem above need not be disjoint - if they are the same color they can share an edge. This theorem is tight in the sense that there are 2-colorings of the edges of K_6 which have only 2 mono-chromatic triangles (both disjoint and non-disjoint cases occur).

Note also that the edges of K_5 can be 2-colored without any mono-chromatic triangles – simply color two disjoint 5-cycles red and blue.

This theorem is a very special case of a much more general theorem called Ramsey's theorem which relates to a topic called **extremal graph theory.** We will we look at one generalization of the problem above in this area.

We can restate the problem as a general question about complete graphs. What is the smallest number n = R(m) such that any 2-coloring of the edges of K_n must have a mono-chromatic sub-graph which contains K_m? We proved above that R(3) = 6, since we showed that any 2-coloring of K_6 must contain a mono-chromatic sub-graph including K_3 (a triangle), but this is not true for K_5.

The numbers R(m) are known as Ramsey numbers. The fourth Ramsey number R(4) is 18. It is possible to prove bounds on the values of other Ramsey numbers, but for m > 4 the exact value of R(m) remains unknown. A lot of work has been done to narrow down the problem so that it is now known that R(5) must be between 43 and 49 (inclusive), but the exact value is extremely difficult to determine despite years of concerted effort. The known ranges of some other Ramsey numbers are listed below:

m	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
R(m)	6	18	43 to 49	102 to 165	205 to 540	282 to 1870	565 to 6588	798 to 23556

It is possible that with further refinements and/or computational power R(5) may be determined in the foreseeable future, however the value of R(6) and any larger Ramsey numbers would appear to be completely out of reach using current technology. The following quote from Paul Erdos aptly sums up the situation:

> *"Imagine an alien force, vastly more powerful than us landing on Earth and demanding the value of R(5), or they will destroy our planet. In that case we should marshal all our computers and all our mathematicians and attempt to find the value. But suppose, instead, that they asked for R(6), then we should attempt to destroy the aliens"*. - Paul Erdős

Turan's Theorem

Another area of extremal graph theory related to the question above asks what is the largest graph with 6 vertices which does not contain a triangle, i.e. how many edges can such a graph have? In terms of the question above this tells us the size of the largest mono-chromatic sub-graph in a 2-coloring of the edges of K_6 which does not contain a triangle (we know of course that the other mono-chromatic must contain two).

Fortunately this question is much easier to answer, and we can do so in a fairly general fashion. We will prove a result for arbitrary graphs which are triangle-free.

Note that a bipartite graph does not contain a triangle, since it has no odd cycles. If n is even, the complete bipartite graph $K_{n/2,n/2}$ will have $n^2/4$ edges – in our example $K_{3,3}$ is a graph with 6 vertices and 9 edges which does not contain a triangle. We will prove that this is the largest number of edges possible.

Theorem 13: Any graph with n vertices which does not contain a triangle has at most $n^2/4$ edges.

Proof: Let G = (V,E) be a graph with n vertices which does not contain a triangle. Let v be the vertex in G with maximum degree. Let S be the set of vertices adjacent to v, and let T=V-S consist of the remaining vertices in G (including v). None of the vertices in S can be adjacent to each other since G does not contain a triangle, so they must all have degree ≤ |T|. All of the vertices in T must have degree less than |S| = d(v), since v had the maximum degree. If we add up the degrees of all the vertices in each set and divide by 2 we find that the number of edges in G is at most (|S|*|T| + |T|*|S|) / 2 = |S|*|T|. We know that |S|+|T| = n, and the maximum value |S|*|T| can have occurs when |S| is as close to |T| as possible (equal if n is even), but is always less than $(n/2)^2 = n^2/4$.
QED

Thus $K_{3,3}$ is the largest triangle-free graph when n = 6. Note that the complement of $K_{3,3}$ is two disjoint triangles, so in some sense when 2-coloring the edges of K_6, maximizing the size of one mono-chromatic sub-graph pushes all the edges in the other sub-graph into triangles.

As with Ramsey theory, the theorem above can be generalized to address the question, what is the largest graph with n vertices that does not contain K_m. When m is greater than 3, the answer will turn out to be a complete (m-1)-partite graph, i.e a graph whose vertices are partitioned into m-1 subsets where no edges occur within a subset. Graphs which maximize the number of edges subject to this constraint are called Turan graphs.