Lecture 7: Bipartite graphs; Matchings

We'll start with a word about the homework, specifically problem 4 (the tripartite graph). If we divide the $n$ vertices into 3 classes, Red, Blue and Green, with Red having $a$ vertices, Blue having $b$ vertices and Green having $c$ vertices, $n = a + b + c$. The maximum number of edges then is $ab + ac + bc$. To calculate the maximum number of vertices as a function of $n$, you need the following:

**Lemma 1.** Given positive numbers $a, b, c > 0$,

$$ab + bc + ca \leq \frac{1}{3}(a + b + c)^2,$$

with equality achieved exactly when $a = b = c$.

**Proof.** Multiplying both sides by 3 and expanding the RHS we get

$$3ab + 3bc + 3ca \leq a^2 + b^2 + c^2 + 2ab + 2bc + 2ca.$$

It is then enough to prove that

$$ab + bc + ca \leq a^2 + b^2 + c^2,$$

or equivalently that

$$2a^2 + 2b^2 + 2c^2 - 2ab - 2bc - 2ca \geq 0.$$

Note that reorganizing and grouping the above we get

$$(a - b)^2 + (b - c)^2 + (c - a)^2 \geq 0,$$

which is certainly true.

What can we conclude? From the above, it follows that the maximum number of vertices is achieved (when possible) when $a = b = c = n/3$. But what about when $n$ is not divisible by 3? A quick inspection should show that in that case we still want to have $a$, $b$, and $c$ as close to each other as possible. You will have to find the exact expressions for this maximum for $n = 3k$, $n = 3k + 1$, and $n = 3k + 2$.

Back to bipartite graphs and maximal numbers of edges. The following theorem should be a bit surprising.

**Theorem 1.** If $H$ is a simple graph on $2m$ vertices with more than $m^2 + 1$ edges ($m \geq 2$), then $H$ contains a triangle.

Note the specificity—a triangle, not just any old odd-length cycle! We prove this by induction.
Proof. The base case is very easy—\( m = 2 \) means that the graph has 4 vertices and at least 5 edges, so a square with one diagonal, hence at least one triangle.

Assume now that we know the result for \( m - 1 \), and we will show it for \( m \).

Pick two connected vertices \( u \) and \( v \) in a graph with \( 2m \) vertices and at least \( m^2 + 1 \) edges. If the sum of their degrees is at least \( 2m + 1 \), it follows that the two vertices have a neighbor \( w \) in common. This is true since the edge between \( u \) and \( v \) contributes 2 to the sum of the degrees, which means that \( u \) and \( v \) have at least \( 2m - 1 \) edges connecting to other vertices; but since there are only \( 2m - 2 \) distinct other vertices, by Pigeonhole, some vertex \( w \) connects to both. Hence \( w \) is a common neighbor of \( u \) and \( v \).

Else, if the sum of their degrees is less than \( 2m + 1 \), that is, at most \( 2m \), the number of edges emanating from \( u \) and \( v \) is at most \( 2m - 1 \) (since in the sum of the degrees the edge connecting \( u \) and \( v \) counts twice).

Thus, deleting the two vertices and all edges adjacent to them creates a graph with \( 2m - 2 = 2(m - 1) \) vertices with at least \( m^2 + 1 - (2m - 1) = (m - 1)^2 + 1 \) edges, at which point induction kicks in to prove that this resulting graph has a triangle. This finishes the proof.

In fact, more is knowable, and I invite you to read the proof for the following theorem from the book (Thm. 11.8).

**Theorem 2.** If \( H \) is a simple graph on \( 2m \) vertices with more than \( m^2 + 1 \) edges (\( m \geq 2 \)), \( H \) has at least \( m \) triangles.

**Matchings.** Suppose one has two sets of objects (e.g., job applicants and positions offered) and one knows exactly which jobs the applicants are qualified for. We are interested in the following question: is there a way to assign to each position a qualified applicant so that all positions are filled?

To be clear, this problem can be modeled by a bipartite graph (\( m \) vertices corresponding to jobs, \( n \) to applicants) with an edge between vertices if and only if the applicant is qualified for the job. Is there a collection of \( m \) vertex-disjoint edges such that each “job” vertex is incident to exactly one of them? That would be a solution to the jobs/applicants problem.

More generally, we give the two definitions below.

**Definition** Let \( G \) be a graph (bipartite or not). A matching is a collection of vertex-disjoint edges. Alternately, this is called a set of independent edges. If every vertex is covered by one of these edges, it is a perfect matching. For a bipartite graph consisting of partition \( (X, Y) \), if a matching covers all the vertices in \( X \), it is a perfect matching of \( X \) into \( Y \).