APM 4663/5663—Fall 2024
Hints for Assignment #1

1. Assume you have k vertices on one side of the bipartition, and find the maximum number of edges the simple graph can have, then find the maximum of this expression. Make sure to justify why you can round down the result and still have a valid inequality. Induction may also work if you go from n to $n+2$ and check two base cases. To show that the result is sharp, consider n even or odd separately, and identify a graph that achieves the maximum number of edges (remember that you need to provide a graph for each possible value of n). For odd n check that you actually get as many edges as given by the formula.

2. Use proof by contradiction, and estimate how many vertices the graph must in each component to satisfy the degree constraints, and count the total number of vertices to get a contradiction. Alternatively, you can show that there is a path between any two vertices after the deletion of any vertex. To show that the result is sharp, again you want to consider n odd or even separately and describe a graph (for each n) that has the necessary properties.

3. Consider a longest path in the graph (or build one as long as possible) and look at the neighbors of one of the ends of this path. This should give you cycles; the longest one of which should be sufficiently long. Make sure to explain why. To show that the result is sharp, identify a graph for each δ with the required properties.

4. Use proof by contradiction, and try to find a path that is longer than the maximum length paths to get a contradiction. Make sure to have a path so you don't repeat a vertex or edge on the walk you find.

5. First try to find a decomposition of K_7 (should be easy) and K_9 (this may be a little trickier). Try to find such a decomposition for K_5 and K_6 as well and see what goes wrong. Consider what must be true for the degrees and the number of edges in K_n to have the required decomposition. Then show that only those n that have the given properties will satisfy these conditions. Note that you don't have to find an actual decomposition for larger n.

6. Start with an arbitrary split of the vertices into two parts, and consider the bipartite subgraph of G with that partition and as many edges as possible (which edges of G can you keep?). If this subgraph does not satisfy the degree conditions, what must happen with a vertex? Show that you can fix this vertex by moving it to the other side and adjusting the edges you keep from G (make sure to describe them). This gives you an algorithm, and show that if it terminates, it will find an appropriate H. To show that the algorithm terminates, consider what happens to the number of edges in each iteration of the algorithm.

   Alternatively, consider a graph H that is a bipartite spanning subgraph of G and has the maximum number of edges under these conditions, then show that H will satisfy the degree conditions.