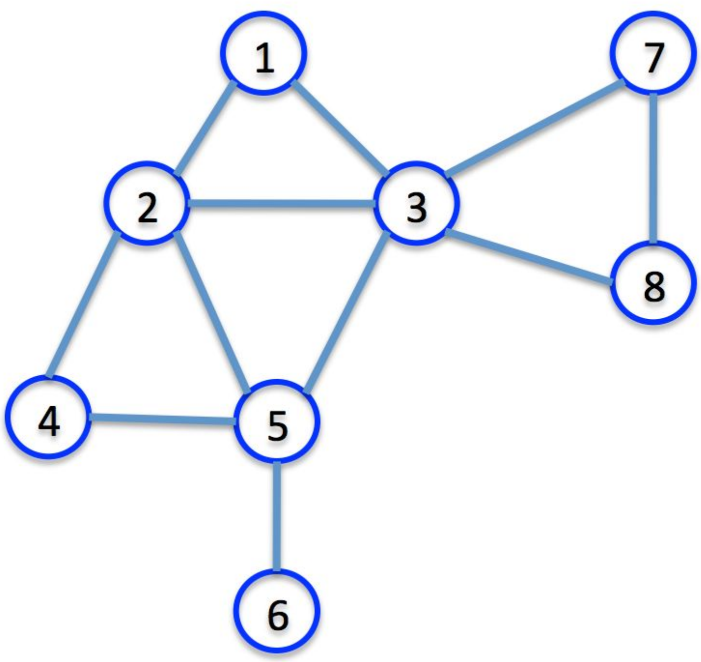
Recitation 6 (10/10 - 10/14)

Announcements:
Next week's recitations will be held like group office hours in preparation for the midterm. If you plan on coming to recitation, come with questions.

Cycles

Talk about valid cycles in the graph below.
Must have at least 3 nodes to be a cycle.



Examples (as output for Q1 will be formatted):

[3,2,5]

[1,2,3]

[5,3,2]

Testing Bipartiteness in Graphs

Definition from Kleinberg-Tardos: Given a graph G, the node set V can be partitioned into sets X and Y in such a way that every edge has one end in X and one end in Y.
We can think of this by coloring each node either red or blue and if there exists an edge such that the nodes on either side are the same color, then the graph is not bipartite.

Q1: Can a triangle cycle be bipartite? Why or why not?

No

Follow up: Can any odd cycle be bipartite?

No

Designing an algorithm to test bipartiteness:

Follow the procedure from the book but have students try and come up with the algorithm.

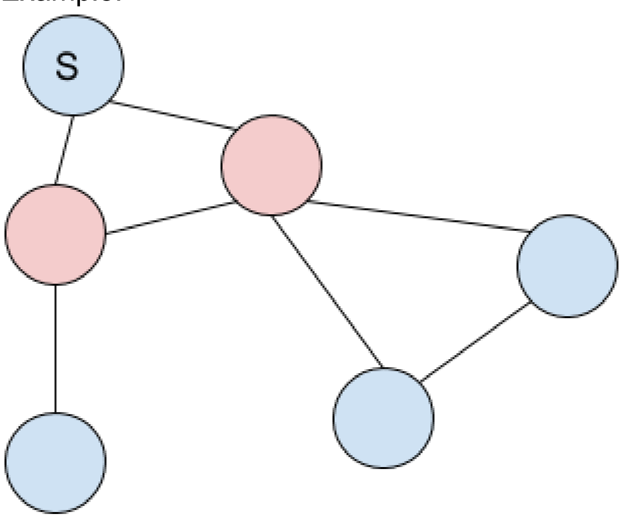
Algorithm:

choose any node s and color it blue
color all of s's neighbors red
color the red node's neighbors blue… etc.

At the end, if every edge has different color nodes on either side, we have a valid bipartite graph.

Talk about how this can be implemented by BFS.

Example:



Not bipartite since we have 2 red nodes on either side of an edge and 2 blue nodes on either side of an edge.

Q3:

Go over the n^3 solution for listing the triangles. The obvious answer is correct. I am not going to write out the actual algorithm in the notes, as that would be free points on the homework. Thanks for checking out the recitation notes! Here is a hint.

Hint: For something to be a triangle, need to check that for nodes u,w,v there exists an edge between u & w, w & v, and u & v. If we can do this check on every node in the tree, then we have listed all of the possible triangles.

Extra time: midterm review.