Graph Theory Problem Ideas

January 19, 2017

Note: Please let me know if you have a problem that you would like me to add to the list!

1 Classification

Given a degree sequence $d_1, \ldots, d_n$, let $N_{d_1, \ldots, d_n}$ denote the number of isomorphism classes of graphs with that degree sequence. Find a formula for $N_{d_1, \ldots, d_n}$. Find an algorithm that computes a graph in each isomorphism class.

Example. In the case of the degree sequence 2, 2, 2, 2, 2, 2, 2, 2, 2, the four different isomorphism classes are drawn at the top of page 65. We could write this as $N_{2^9} = 4$. Similarly, if you solve Exercise 3.16, you should realize that $N_{6^9} = 4$ (hint: use Theorem 3.1). What is the symmetry at work here? Can you generalize this to get an identity for $N_{d_1, \ldots, d_n}$?

Remarks.
- Start with simple degree sequences! For example, compute $N_{1^n}$ and $N_{2^n}$.
- Solving this problem for $N_{r^n}$ ($r$-regular graphs with $n$ vertices) would already be extremely interesting.
- Ideally, you would find a closed formula for $N_{d_1, \ldots, d_n}$ in terms of the variables $d_1, \ldots, d_n$. If a closed formula proves to be elusive, it may be possible to show that the numbers $N_{d_1, \ldots, d_n}$ for various choices of degree sequences can be assembled into a nice structure, such as a power series that has a nice factorization.
- One approach to computing $N_{d_1, \ldots, d_n}$ is to find relationships between these numbers as you vary the degree sequences. For instance, you could investigate how the number changes when you subtract 1 from $d_1$.
- Sometimes small changes to a counting problem can lead to a much nicer formula. For instance, counting all pseudographs with a given degree sequence or counting only connected graphs with a given degree sequence may be easier.

2 Checking isomorphism

Devise an efficient algorithm to determine whether two graphs are isomorphic.
Remarks. • The brute-force algorithm we saw in class takes $n!$ time, which is considered to be extremely bad. You should aim for a polynomial-time algorithm (e.g. $n^3$).

• Once again, restricting to a special class of graphs (such as $r$-regular graphs or connected graphs) may make this problem more tractable.

3 Graphical degree sequences

Devise a list of conditions for degree sequences such that every degree sequence satisfying those conditions is graphical.

Remark. We already have an efficient (linear time) algorithm for determining whether a sequence is graphical. You could try to use this algorithm to help you devise the list of conditions.

4 Properties of a graph and its adjacency matrix

How are properties of a graph (such as the degree sequence, connectedness, bipartiteness, etc.) reflected in the adjacency matrix? Conversely, how are properties of the adjacency matrix (such as invertibility, eigenvalues, eigenvectors, etc.) reflected in the associated graph?