MAT 331 Project
Perfect matchings

Suppose G is a bipartite graph of size $2N$ and let V_1, V_2 be the two subsets of vertices, each of size N, so that edges only connect V_1 to V_2 and conversely (no vertex of V_1 is connected to another vertex in V_1 and the same for V_2). We will assume V_1 has vertices $1, \ldots, N$ and V_2 has vertices $N+1, \ldots, 2N$.

A perfect matching is a set of N edges in G that uses every vertex exactly once; each vertex in V is paired with one vertex in V and none are left out.

Tutte's theorem says that a perfect matching exists if and only if the determinate of a certain matrix is a non-zero polynomial. The matrix is a $2N \times 2N$ matrix that has a variable x_{ij} in position (i, j) if $i < j$ and v_i is connected to v_j, has $-x_{ij}$ in position (i, j) if $i > j$ and v_i is connected to v_j and is zero everywhere else. The determinate of this matrix is a polynomial in these variables and Tutte's theorem says it is zero iff there is no perfect matching (so there is a matching iff it is not the zero polynomial).

It's hard to compute this polynomial, but it is easy to plug in random values for the variables x_{ij}. If we do this and keep getting zero every time, then we can be sure the polynomial is zero with high probability.

(1) Write a script that inputs N and creates a bipartite matrix that has a perfect matching. Start with connecting each vertex k to $k + N$ for $k = 1, \ldots, n$. Then add other edges at random to the connect V_1 V_2.
(2) Write a script that inputs N and creates a bipartite matrix that does not have a perfect matching. You can do this by creating a random bipartite graph and then some vertex is not connected to any other vertex. Can you think of some other way to do this?
(3) Write a script that uses Tutte's determinate test to decide if an input graph has a perfect matching or not. Try it on the graphs built in parts (1) and (2) and report your results.
(4) Write a script that finds a perfect matching as follows. Start by applying Tutte's test. If there is no perfect matching, the program should report 'no perfect matching'. If Tutte's test says there is a perfect matching, then choose an edge in the graph and remove it from the graph and apply Tutte's test to the new graph. If the new graph still has a perfect matching, and store the removed edge and continue by removing another edge. If the new graph does not have a perfect matching, put the edge back (and mark it so you do not remove it again), and try removing a different, unmarked edge. Repeat this until you have found a perfect matching.
(5) Test your code using the graphs produced in parts (1) and (2). Report your results.