Your Name	November 6, 2026
6.5240 Problem Set 5	
Collaborators: *Firstname Lastname*	Due: **10pm**, November 18, 2026

Homework guidelines: You may work with other students, as long as (1) they have not yet solved the problem, (2) you write down the names of all other students with which you discussed the problem, and (3) you write up the solution on your own. No points will be deducted, no matter how many people you talk to, as long as you are honest. If you already knew the answer to one of the problems (call these "famous" problems), then let us know that in your solution writeup – it will not affect your score, but will help us in the future. It's ok to look up famous sums and inequalities that help you to solve the problem, but don't look up an entire solution.

1. **Moore Bound** In this question, we will develop Moore Bound: For every $k \geq 2$, simple undirected graphs of n vertices with girth[1] at least $2k+1$ can have only $O(n^{1+1/k})$ edges.

   (a) For a graph G of n vertices, m edges and girth at least $2k+1$, define $\bar{d} = 2m/n$ to be its average degree. Show that removing a vertex in G with degree smaller than $\bar{d}/2$ along with all its incident edges will increase the average degree of G.

   (b) From (a), conclude that every graph $G(V, E)$ has nonempty $U \subseteq V$ such that induced subgraph $G' = G[U]$[2] has minimum degree $\bar{d}/2$ and girth at least $2k+1$.

   (c) Following (b), show that for every vertex v in G', the set of vertices with distance at most k from v in G' is at least $\sum_{i=0}^{k}(\bar{d}/2-1)^i$.

   (d) Using (c) and the fact that G' has at most n vertices, show that $m = O(n^{1+1/k})$.

2. **Sublinear-time 2-coloring for Random Bipartite Graphs** In this question, we will consider a sublinear-time algorithm for 2-coloring in the adjacency matrix model. The algorithm will succeed with high probability over a random graph generated by the following process.

   - Given the number of vertices n (we assume n is even), the algorithm generates a equal bipartition V_1, V_2 of the vertex set V uniformly from all possible equal bipartitions.
   - An edge in the graph always connects one vertex in V_1 and one vertex in V_2. For every $u \in V_1$ and $v \in V_2$, we flip a fair coin, and the edge (u, v) belongs to the graph when the coin flips heads.

[1] The *girth* of a graph is the minimum length among all simple cycle in the graph.

[2] For a graph $G(V, E)$ and $U \subseteq V$, the *induced subgraph* $G[U]$ is the graph with vertex set U, and an edge belongs to $G[U]$ if and only if it belongs to E and both endpoints belong to U.

All randomness in the process above is mutually independent. One can see that every graph generated by the process admits a 2-coloring.

(a) The first step is to find an edge (u, v) in the graph. Show that this can be done within $O(\log n)$ queries with probability at least $1 - O(1/n^2)$.

(b) Given an edge (u, v) in the graph, we color the vertices with oracle $\mathcal{O}_1$. Given a vertex w, $\mathcal{O}_1$ colors w with red when $w = u$ and colors w with blue when $w = v$. For other vertices, it

- queries (u, w) and (v, w),
- color w with blue if (u, w) is in the graph,
- color w with red if (v, w) is in the graph,
- and postpone the coloring for w if neither (u, w) nor (v, w) is in the graph.

Show that $\mathcal{O}_1$ colors at least $n/6$ vertices with red and at least $n/6$ vertices with blue with probability $1 - O(1/n^2)$.

(c) Finally, given a vertex w, consider the following oracle $\mathcal{O}_2$ to color the vertices. When $\mathcal{O}_1(w)$ gives w a color, $\mathcal{O}_2(w)$ gives the same color. Otherwise, it repeats the following procedure $O(\log n)$ times, outputs the corresponding color of any of them gives w a color, and output FAIL otherwise:

- Uniformly sample a vertex x.
- If $(w, x) \in E$ and $\mathcal{O}_1(x)$ gives x a color, give the other color to w.

Show that, with probability $1 - O(1/n)$, $\mathcal{O}_2(w)$ never outputs FAIL for every vertex w, and the coloring given by oracle $\mathcal{O}_2$ is a valid 2-coloring.