HW 2

Due: September 11th, 2025

Problem 1: Hardness for CLIQUE

Recall that a CLIQUE in an undirected graph $G = (V, E)$ is defined as a subset of vertices $C \subseteq V$ such that every vertex $v \in C$ has an edge to all other vertices $u \in C$. The definition of the maximum CLIQUE problem from class is: Given an undirected graph with n vertices $G = (V, E)$ find a clique C of maximum size. Let $\omega(G)$ be the size of the largest CLIQUE.

In class we gave this as an example of an NP-complete problem. In particular, even deciding whether $\omega(G) \geq k$ for given G and k is NP-complete.

Now, we will look at an approximate version of this problem where the goal is to find a CLIQUE that is within a factor of α of the maximum size, for some constant $0 < \alpha < 1$.

To do this we will construct a new graph $G^{(r)}$ which is defined as:

- The vertex set of $G^{(r)}$ is $V^r = V \times V \times V \times \ldots \times V$ (r times).
- Two vertices $(v_1, \ldots, v_r)$ and $(u_1, \ldots, u_r)$ in $G^{(r)}$ are adjacent if and only if for all $i = 1, \ldots, r$ either $v_i = u_i$ or $(v_i, u_i) \in E$

Part A.

Prove that if G has a CLIQUE of size k, then $G^{(r)}$ has a CLIQUE of size k^r

Part B.

Prove that if G does not have a CLIQUE of size s, then $G^{(r)}$ has no CLIQUE of size s^r

Part C.

Suppose we had a polynomial-time algorithm A that, given any graph H, returns a clique of size at least $\alpha \cdot \omega(H)$, for some constant $\alpha > 0$. Show how, for *any* given constant $\beta < 1$, we can use this algorithm on some graph $G^{(r)}$ to distinguish between the following two cases in polynomial time:

- **(YES)** $\omega(G) \geq k$
- **(NO)** $\omega(G) < \beta \cdot k$

By choosing r appropriately, explain how the gap between k^r and s^r grows so that the approximation algorithm would be able to distinguish them. (Ultimately this leads to a contradiction unless $P = NP$, but you don't need to show this part.)

(Hint: Use the results from the first two parts to understand how the maximum clique size behaves during the transformation from G to $G^{(r)}$.)

Problem 2:

You are given a model of a distributed system where agents contend for resources and back off in the face of competition. We're going to use an abstract model where balls represent agents and bins represent resources.

The system evolves over rounds. In the first part of every round, balls are thrown independently and uniformly at random into n bins. In the second part of each round, each bin, in which at least one ball has landed in that round, serves exactly one ball from that round. The remaining balls are thrown again in the next round. We begin with n balls in the first round, and finish when all balls are served.

Part A.

Show that, with probability $1 - O(1/n)$, all balls are served within $\log_2 \log_2 n + O(1)$ rounds.
(Hint: Let b_k be the number of balls left after k rounds; show that $b_{k+1} \leq c(b_k)^2/n$, for a suitable constant c with high probability, as long as b_{k+1} is sufficiently large.)

Part B.

Suppose that we modify the system so that a bin accepts a ball in a round if and only if that ball was the *only* ball to request that bin in that round. Show that, again with probability $1-O(1/n)$, this approach takes at most $\log_2 \log_2 n+O(1)$ rounds.