PROBLEM SET 7

Zhiqiang Xie 77892769

Matrix Fixing

1. Can we set a $n \times n$ matrix M of integers into a matrix of all zeros in k operations?
2. Given a sequence of operations with the lenght of k, we can execute them and check whether all entries are zero or not in $T(n) = O(k * n) + O(n^2) \in O(n^2)$. Thus, MATRIX-FIXING $\in$ NP.
3. CLIQUE $\leq_p$ MATRIX-FIXING
   - Transformation:
     - Given an arbitrary instance of CLIQUE, $G = (V, E, m)$ where m means a $m-$vertex clique. We construct the instance of MATRIX-FIXING, $F(M, k)$ where $k = |V| - m$ and M is a $|V| \times |V|$ matrix. Then we set the elements of the matrix:
       - $M_{i,i} = 0$ for all $i \in [1, |V|]$
       - $M_{i,j} = 0$ for all $(i, j) \in E$. Note here G is an undirected graph and $(i, j) = (j, i)$
       - $M_{i,j} = 1$ for any entries else.

     The transformation cost $O(n^2)$ time.
   - Proof:

     We now need to prove that G is a yes-instance of CLIQUE if and only if that $F(M, k)$ is a yes-instance of MATRIX-FIXING.

     - ($\Rightarrow$) Suppose that G is a yes instance, thus there are at least m rows and columns of all zeros. Therefore, we can set the residues to be all zeros in $|V| - m = k$ operations.
     - ($\Leftarrow$) Suppose that $F(M, k)$ is a yes-instance. Since every operation (set the $i'th$ row abd the column to zero) is equivalent to connect the corresponding vertex in the graph to any other vertices. Once we can set all entries to be zero we can create a $|V|-$vertex clique by adding at most k vertices to the original clique. Therefore, there must be a $m-$vertex clique inside the original instance G. Thus, it's a yes-instance.

   MATRIX-FIXING problem is NP-complete.
4. Suppose that there is no operation which can change $\frac{1}{k}$ fraction, thus for all operation $\frac{1}{k_i} < \frac{1}{k}$.
   - Therefore, for the optimal k operations, we have $\sum_{i=1}^{k} \frac{1}{k_i} < k * \frac{1}{k} = 1$
   - At that case, no operation sequence with length k can solve this problem.
5. Since we already know that we can find a operation which can change at least $\frac{1}{k}$ fraction of the nonzero entries by greedy method.
   - Thus, here are at most $(1 - \frac{1}{k})$ fraction of nonezero entries left.
   - Every time we do this operation, we can concatenate the residue submatrixs into one with the size of $(n-1) \times (n-1)$. For the new matrix, the optimal number of operations is no more than k.
   - After t greedy operations, we have at most $(1 - \frac{1}{k})^t$ fraction of the nonezero entries. Let $t = 2k \log n$, we have $n^2 * (1 - \frac{1}{k})^{2k \log n} < n^2 * e^{-\frac{1}{k} * 2k \log n} = 1$. Hence, all entries will be zero in at most $2k \log n$ operations.

TRIPLE-SAT

1. TRIPLE-SAT ∈ NP
   - Given an arbitary solution (Here the solution should be three distinct assignments) to TRIPLE-SAT, we can evaluate the Boolean formula in polynomial time by basic logical operations.
2. 3-SAT $\leq_p$ TRIPLE-SAT
   - Transformation:
     - Given an arbitary instance of 3-SAT, a 3-CNF formula f. We construct the instance of TRIPLE-SAT, the Boolean formula $\phi = f \wedge (x \vee y)$. Where x and y are two variables. This step cost $O(1)$.
   - Proof:

     We now need to prove that f is a yes-instance of 3-SAT if and only if that ϕ is a yes-instance of TRIPLE-SAT.

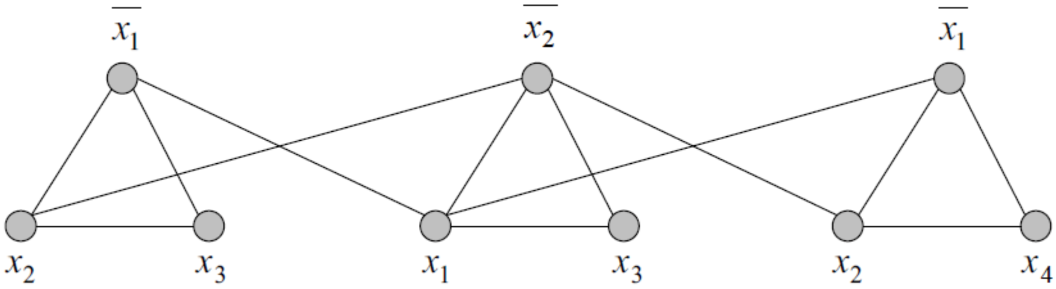
     - ($\Rightarrow$) Suppose that f is a yes-instance, we can assign these three assignments to (x, y) to make ϕ a satisfiable formula.
       - $(T, F), (F, T), (T, T)$

       Therefore, ϕ is a yes-instance of TRIPLE-SAT.
     - ($\Leftarrow$) Suppose that ϕ is a yes-instance, it requires that f to be satisfiable, or ϕ will never be a yes-instance. Thus f is a yes-instance of 3-SAT.

   TRIPLE-SAT is NP-complete.

BAGEL

1. BAGEL ∈ NP
   - Given an arbitary solution to BAGEL, we can sum out the result and compare it to k in $O(n)$ time.
2. 3-SAT $\leq_p$ BAGEL
   - Transformation:
     - Given an arbitary instance of 3-SAT, a 3-CNF formula $f(k)$, where k is the number of clause. We construct a instance of BAGEL $B = (G, p, k)$, where $G = (V, E)$ and $|V| = 3 * k, p(u) = 1$ for all $u \in V$.
     - G contains 3 vertices for each clause, one for each literal.
     - Connect 3 literals in a clause in a triangle.
     - Connect literal to each of its negations.



$$\Phi = (\overline{x_1} \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee x_4)$$

This transformation costs $O(n)$ time.

- Proof:

  We now need to prove that $f(k)$ is a yes-instance of 3-SAT if and only if B is a yes-instance of BAGEL.

  - (⇒) Suppose that $f(k)$ is a yes-instance, here exists a satisfying assignment which select exact one true literal from each clause. Hence it's a subset $U \subseteq V$ such that no two vertices in U are neighbors in G, and $\sum_{u \in U} p(u) = k$. Here B is a yes-instance.
  - (⇐) Suppose that B is a yes-instance, only one vertex in each triangle can be in the subset U. Since all vertices in U are not adjacent. We can assign all the literals in U to be true, and the assignment indicates that $f(k)$ is satisfiable.

BAGEL is NP-complete.