CME305 Sample Midterm II

1. **Matchings and Vertex Covers**

   (a) Define what a matching in G is.

   (b) Define what a vertex cover of G is.

   (c) Let M be a maximum matching and C a minimum vertex cover. Show that $|M| \leq |C| \leq 2|M|$.

   Solution:

   1. A matching M is a subset of E such that no two edges share an endpoint.
   2. A vertex cover C is a subset of V such that all edges are incident to at least one element of C.
   3. Let M be a maximum matching and C a minimum vertex cover. It is easy to see that we need at least $|M|$ nodes to cover all the edges of M. Hence $|M| \leq |C|$. Now let us prove that $V(M)$, the vertices in the matching M, is a vertex cover. Assume it isn't, then there is at least one edge e that is not covered by $V(M)$. It is easy to see that this implies that $M \cup \{e\}$ is a matching so M is not maximum. Contradiction. Furthermore, we know that $|V(M)| = 2|M|$, hence $|C| \leq 2|M|$.

2. **Traveling Salesman Problem**

   Assume that deciding whether a graph has a Hamiltonian cycle is NP-Complete. Prove that the Traveling Salesman Problem is NP-Hard.

   Solution: As defined in class the TSP problem defines a complete graph K_n with a cost function $c : E \rightarrow \Re^+$ and asks to find a cycle that visits all vertices exactly once and such that the cost of the cycle is minimized.

   We are now going to show how to solve Hamiltonian Cycle on G by using TSP and a polynomial amount of work. For this, we will use the decision version of TSP, which is "is there a TSP of cost at most k?".

We reduce Hamiltonian cycle HC to TSP i.e. show $HC \leq_P TSP$. Given $G(V, E)$ with $|V| = n$ we define $c(e) = 1$ for all $e \in E$. Then we add edges E' to G to make G a complete graph and assign $c(e) = 2$ for all $e \in E'$. We can do this in polynomial time. Now given this cost, if the answer to is there a TSP cycle of cost at most n is "yes", then we know there is a cycle that visits all nodes exactly once. The edges it uses are from E (given the cost function we created), hence we can say for sure that there is a Hamiltonian Cycle in G. Given that Hamiltonian Cycle is NP-Complete, the decision version of TSP is NP-Complete, hence TSP is NP-Hard.

3. **Lecture Attendance Planning**

A group of students want to minimize their lecture attendance by sending only one of the group to each of the n lectures. We have the following constraints:

- Each of the n lectures should be covered.
- Lecture i starts at time a_i and ends at time b_i.
- It takes r_{ij} time to commute from lecture i to lecture j.
- Assume all times r_{ij} as well as the duration of the lectures are in minutes and integers.

Minimize the number of students that will attend lectures i.e. develop a flow based algorithm to identify the minimum number of students needed to cover all n lectures.

Solution: We are going to solve this problem using a maximum matching on a bipartite graph. First observe that minimizing the number of students attending courses is equivalent to maximizing the number of classes that a given student can attend. If classes i and j are such that one can go to class i, commute to class j and still be on-time, then we only need one student to go to both classes. Let us now build the graph:

For each lecture i, set two nodes x_i and y_i. $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_n\}$ are our partitions. The edge (x_i, y_j) exists if $i \neq j$ and one can go to class i and then to class j. More formally $i \neq j$ and $a_j \geq b_i + r_{ij}$.

Building such graph takes at most $2n \times 3n^2$ steps, polynomial in n.

Let M be a maximum matching in $G(X, Y, E)$. We claim that the minimum number of students needed is $n - |M|$. We can prove this by contradiction.

Assume you can go to all the lectures with $n-p < n-|M|$ students. Then that means that we can "reuse" p students. Let $I = \{i_1, \ldots, i_p\}$ be the set of lectures where we are reusing a student (i.e. the set of lectures that at the end the student will go to another lecture). Let $J = \{j_1, \ldots, j_p\}$ be the set of lectures they are attending afterwards. It is easy to see that $M' = \{(x_{i1}, y_{j1}), ..., (x_{ip}, y_{jp})\}$ is a matching in G. But $p > |M|$, which is a contradiction.

We can find such matching using a flow algorithm, just like we saw in class.