Name - Rohan Patel

Student Number - 11247205

NSID - rsp502

**Assignment 5**

**Question 1**

Assume that a problem A cannot be solved in O(n^2) time. However, we can transform A into a problem B in O(n^2 log n) time, and then solve B, and finally transform the solution of B in O(n) time into a solution for A.

->

As we know, when we reduce problem A to problem B, we are indirectly saying that we can solve problem B in a particular time complexity, then we could solve problem A, in the same time complexity. (as long as the reduction is sufficiently simple that it can be performed within the relevant time complexity). Which means that B is at least as hard as A. Problem A can be simpler than problem B. For instance, let problem A ∈ **P** and let B be any problem that can be solved in exponential time. Because B is exponential time problem, every problem in exponential time reduces to B so, in particular, A reduces to B. But we know that A is strictly easier than B by the time hierarchy theorem.

So, we can say that as we cannot solve A in O(n^2) and Problem B can be as hard as problem A or harder. It can’t be solved in O(n^2).

**Question 2**

Prove that Minimum Vertex Cover problem is NP-hard by reducing the NP-hard problem Maximum Independent Set.

->

A Maximum independent set is an independent set of largest possible size for a graph and Vertex Cover is a graph G and a number k, does G contain a vertex cover of size at most k. So, If G = (V, E) is a graph, then S is an independent set ⇐⇒ V − S is a vertex cover.

Proof.

Suppose S is an independent set, and let e = (u, v) be some edge. Only one of u, v can be in S. Hence, at least one of u, v is in V − S. So, V − S is a vertex cover.

Suppose V − S is a vertex cover, and let u, v ∈ S. There can’t be an edge between u and v (otherwise, that edge wouldn’t be covered in V − S). So, S is an independent set.

Independent Set ≤ P Vertex Cover

To show this, we change any instance of Independent Set into an instance of Vertex Cover:

• Given an instance of Independent Set (G, k),

• Vertex Cover black box if there is a vertex cover V − S of size ≤ |V| − k.

S is an independent set if V − S is a vertex cover. If the Vertex Cover black box said:

If yes, then S must be an independent set of size ≥ k.

If no, then there is no vertex cover V − S of size ≤ |V| − k, hence there is no independent set of size ≥ k.

We also know that Vertex Cover ≤ P Independent Set.

To decide if G has a vertex cover of size k, we ask if it has an independent set of size n − k.

So, we can say that Vertex Cover and Independent Set are equivalently difficult.

The Maximum Independent Set problem consists in computing an Independent Set of largest size that is the largest set of vertices that induces no edges. The problem can be generalized to the case when a weight is associated with each vertex. Let the Minimum Vertex Cover be the problem of finding the complement of this set (i.e. the smallest set of vertices that touch all edges). Clearly, for every graph G, a solution to Minimum Vertex Cover is the complement of the solution to Maximum Independent Set. Thus, we can deduce that minimum vertex cover problem is NP-hard by reducing the maximum independent set problem.

**Question 3**

Since finding a minimum vertex cover in a graph is known to be NP-hard, we want to

find a vertex cover S that is not too large than a minimum vertex cover. We propose the

following algorithm.

Step 1. Initialize S with an empty set.

Step 2. While there is an edge in G, randomly choose an edge (a, b) and insert a and b into

S. Then delete the vertices a and b, and all the edges incident to a and b.

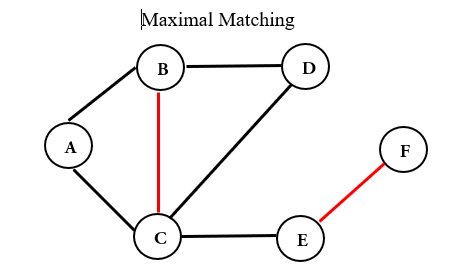
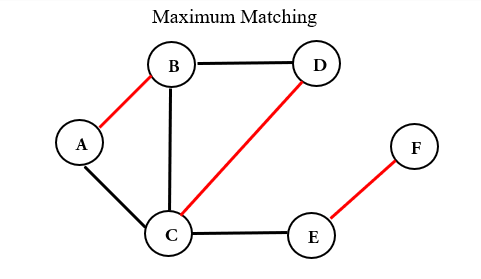
Step 3. Return S

Prove that the size of the set S returned by the algorithm can be at most twice the size of

a minimum vertex cover of G.

->

What the algorithm finds in the end is a maximal matching which means, a set of edges no two of which share an endpoint. We can notice that a maximum matching is also a maximal matching, but the opposite is not necessarily true. See the following example:



Twice the size of a minimum vertex cover of G. Let 𝑀 be a maximal matching, and 𝑆 be the set of all endpoints in 𝑀.

Proof

We prove this by contradiction: suppose there exists an edge 𝑒 = (𝑢, 𝑣) such that 𝑢, 𝑣 ∉ 𝑆. Since 𝑒 does not share an endpoint with any of the vertices in 𝑀, 𝑀 ∪ {𝑒} is a larger matching, which contradicts 𝑀 being a maximal matching.

Given algorithm gives a 2-approximation for minimum vertex cover regardless of the choice of *𝑀*.

Let 𝑂𝑃𝑇 be the minimum vertex cover in graph 𝐺. The edges of 𝑀 are independent; thus, any feasible cover must take at least one vertex from every edge in 𝑀. This means that |𝑀|≤𝑂𝑃𝑇 and then we have:

|𝑀| ≤ 𝑂𝑃𝑇 ≤ |𝑆| = 2|𝑀| ≤ 2𝑂𝑃𝑇

This technique of lower bounding the optimum is often key in proving approximation factors, as we are usually unable to compute the value of OPT.

So we can deduce, that the size of set S i.e the size of vertex returned can be double the size of minimum size vertex cover of G.

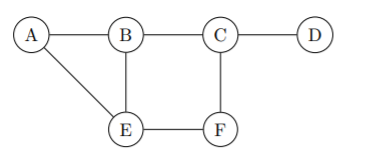
**Question 4**

You are given a social network of n students, where two students are connected if and only if they are friends. Consider the problem of finding a smallest set S of students from the network such that any student in the network is either in S, or a friend of someone who belongs to S.

Either give a polynomial-time algorithm for the problem or show that the problem is NP-hard by reducing the Minimum Vertex Cover problem.

->

In the minimum vertex cover problem, we are given an undirected graph G = (V, E) and asked to find the smallest set U ⊆ V that “covers” the set of edges E. In other words, we want to find the smallest set U such that for each (u, v) ∈ E, either u or v is in U (U is not necessarily unique). For example, in the following graph, {A, E, C, D} is a vertex cover, but not a minimum vertex cover. The minimum vertex covers are {B, E, C} and {A, E, C}.



Recall the following definition of the minimum Set Cover problem: Given a set U of elements and a collection S1, . . . , Sm of subsets of U, what is the smallest collection of these sets whose union equals U? So, for example, given U := {a, b, c, d}, S1 := {a, b, c}, S2 := {b, c}, and S3 := {c, d}, a solution to the problem is the collection of S1 and S3.

Give an efficient reduction from the Minimum Vertex Cover Problem to the Minimum Set Cover Problem.

Proof:

Let G = (V, E) be an instance of the Minimum Vertex Cover Problem. Create an instance of the Minimum Set Cover Problem where U = E and for each u ∈ V, the set Su contains all edges adjacent to u. Let C = {Su1, Su2, . . . , Suk} be a set cover. Then our corresponding vertex cover will be u1, u2, . . . , uk. To see this is a vertex cover, take any (u, v) ∈ E. Since (u, v) ∈ U, there is some set Sui containing (u, v), so ui equals u or v and (u, v) is covered in the vertex cover.

Now take any vertex cover u1, . . . , uk. To see that Su1, . . . , Suk is a set cover, take any (u, v) ∈ E. By the definition of vertex cover, there is an i, such that either u = ui or v = ui.

So (u, v) ∈ Sui, so Su1, . . . , Suk is a set cover.

Since every vertex cover has a corresponding set cover (and vice-versa) and minimizing set cover minimizes the corresponding vertex cover, the reduction holds.