Aditya Shirode, Rhythm Shah, Shrenuj Gandhi

2. [2 points for (a), 8 points for (b)]

Define the *graph coloring problem* (optimization version) as follows. Given an undirected graph $G = (V, E)$ find a function $c : V \to \{1, \ldots, k\}$ such that, for every $uv \in E$, $c(u) \neq c(v)$ and k is minimized. In other words, use the minimum number of colors to color the vertices so that the endpoints of every edge have a different color. A famous example of this problem is the four-color map theorem, which says that, if G is planar, then there exists a solution with $k \leq 4$.

(a) Describe a linear time algorithm that determines whether or not G has a two-coloring, i.e., a function c with $k = 2$.

(b) Give a precise definition of a decision version of the graph coloring problem and prove that the decision version can be solved in polynomial time if and only if the optimization version can. The hardest part of this is a self-reduction to show that an algorithm for the decision version can be used to find a certificate (a coloring). Here it is important to note that the actual color of a vertex does not matter – what matters is which vertices have the same color and which ones have different colors.

Answer:

(a) To determine whether an undirected graph G has 2-coloring solution, we implement a modified version of BFS search.
We assign colors (say *red* and *blue*) to alternate layers. Then go over all the edges and check whether the two endpoints of this edge have different colors.

These steps can be done in O($|V| + |E|$), which is linear in the size of the graph.

(b) The decision version of the graph coloring problem is as follows -

Given an undirected graph $G = (V, E)$ and a positive integer k,
does there exist a function $c : V \to 1 \ldots k$,
such that $(u, v) \in E$ and $c(u) \neq c(v)$?

To prove the biconditional statement : decision problem can be solved in polynomial time iff the optimization version can, we break it into two parts.

Part 1: If optimization problem can be solved in polynomial time, then the decision version can be solved in polynomial time.

From the relationship among problem types, we know that solving an optimization problem yields solution to all the others.

Thus having a polynomial time optimization problem that finds k-coloring with minimum k, we can deduce the evaluation problem in polynomial time to return k.
This value of k can be the basis of decision problem, thus solving it in the same polynomial time.

Part 2: If the decision problem can be solved in polynomial time, then the optimization version can be solved in polynomial time.

To prove polynomial time optimization problem, we need two things:
- polynomial time decision algorithm (*given*)
- polynomial time certificate algorithm

Let *GC-DEC(G,k)* be the boolean function that solves the decision problem in polynomial time.
Define $G/uv = (V', E') \in V' = V - v$ and

$$E' = E - \{vw|vw \in E \forall w \in V\} + \{uw|vw \in E \text{ and } uw \notin E \forall w \in V\}.$$

The following is the certificate algorithm -

```
GC–CERT(G, k):
        if |V| ≤ k, then:
                return k–coloring c with different colors for each vertex

        for all pairs (u,v), do:
                if (u,v) ∉ E and GC-DEC(G/uv, k), then:
                        c ← GC-CERT(G/uv, k)
                        c(u) = c(v)
                        return c

        return false
end
```

To compute G/uv, we require $O(n^2)$ time, where $n = |V|$.
Let $D(n)$ be the time polynomial taken by the decision algorithm.
Thus the time taken by certificate algorithm is $n^2(O(n^2) + D(n))$
i.e. $O(n^4 + n^2 D(n))$, which is a polynomial.

Since both decision and certificate algorithms are polynomial time, the optimization problem can also be solved in polynomial time.