Y. Shivananda B220603CS CS04.

Q1) Prove that following are polynomial time veritiable
1) Independent set

Ans) Problem Definition:

An Independent set of a graph $G(V,E)$ is a subset $S \subseteq V$ such that no two vertices in S are adjacent. The problem IND_SET asks whether there exists an independent set of size at least K.

IND_SET = $\{<G,K> \mid G$ is an Encoding of an undirected graph G and K is an Encoding of +ve integer such that G has a independent set of size $\geq K\}$.

Input: Graph G, integer K, $S' \subseteq G.V$

Output: "1" if S' is a independent set
"0" otherwise.

Algorithm:

```
IND_SET_Verification(G,K,S'):
1. if (|S|<K) print 0
2. else   E' = G.E
          reject=0
          for u in S':
             for v in S':
               |-- if (u==v) continue
               |-- edge e = (u,v)
               |-- if (e ∈ E') reject=1
        if (reject==1) print 0
        else print 1.
```

Complexity Analysis: checking $|S'| \geq K \rightarrow O(1)$

checking if any edge exists in S' requires:

- $O(|S'|^2)$ comparisons
- $O(1)$ lookup for each edge in adjacency list (assuming efficient representation)
- Total $O(n^2) \Rightarrow$ Polynomial Time.

Hence we can say that Independent set is polynomial time verifiable.

2) Colouring:

Ans. COLOURING = {<G, K> | G is an Encoding of an undirected graph G and K is an Encoding of +ve integer such that there is a valid vertex colouring of G with C, wher $|C| = K$.}

Input: Graph G, integer K, function $f: V \rightarrow \{1, 2, ---, K\}$

Output: "1" if f is a valid K-colouring, "0" otherwise

Algorithm:

```
colouring_verification (G, K, f):
1. V' = G.V , E' = G.E
2. for v in V'
       if (f(v) < 1 || f(v) > K) return 0
3. for edge (u,v) in E'
       if (f(u) == f(v)) return 0.
```

Complexity Analysis: checking if colors are within range $\rightarrow O(|V|)$

checking edge constraints $\rightarrow O(|E|)$

Total complexity: $O(|V|+|E|)$

$\hookrightarrow$ polynomial time.

Hence, we can say that colouring is polynomial verifiable.

3) CNFSAT:

Ans: CNF_SAT = { F | F is a boolean formula, F in CNF and F is satisfiable }.

Input: Boolean Formula F with m clauses and n variables Assignment A for n variables.

Output: "1" if A is satisfies F, "0" otherwise.

Algorithm:

```
CNF_SAT_Verification (F,A):
1. for each clause C in F
       satisfied = false
     for each literal l in C
     |— if l is satisfied under A
     |     |— satisfied = true
     |     |— break
     if (satisfied == false)
         return 0
   return 1
```

Complexity Analysis:

checking each clause → $O(n)$

checking literals in each clause → $O(n)$ in worst case

Total Time Complexity: $O(mn)$

Hence, we can say that CNF_SAT is polynomial time verifiable.