CSC373H1 Summer 2014 Assignment 4

Names: John Armstrong, Henry Ku
SNs\CDF username: 993114492\g2jarmst, 998551348\g2kuhenr

Question #	Score
1	
2	
3	
4	
Total	

Acknowledgements:

"We declare that we have not used any outside help in completing this assignment."

Name: John Armstrong, Henry Ku Date: July 28, 2014

Q1. The Mute Prison

Claim: The mute prison problem is NP-complete.

Proof:

1. Show the mute prison problem is NP.

2. Show the mute prison problem is NP-hard.

1. Suppose we are given a certificate S and have access to value k and matrix T. We can verify that the certificate is satisfiable in the following way. Suppose each element in S represents an inmate. Verification would involve iterating on each inmate in the following way:

```
for inmate in S do
    j = 1;
    while j ⩽ m do
        if T[inmate, j] then
            for (otherinmate ≠ inmate) in S do
                if T[otherinmate, j] then
                    S is not a subset of inmates who do don't speak the same language;
                end
            end
        end
        j++;
    end
end
```

Clearly, the verification that S is a subset where no two inmates speak the same language can run in polynomial time O(mn^2). Once this verification if complete all that is left to do is to verify that $|S| \geqslant k$, which is O(1). Therefore the mute prison problem is NP. ■

2. To show that the mute prison problem is NP-hard we must perform a reduction using an NP-complete problem. We will use a reduction on NP-complete 3-SAT in CNF, in order to show 3-SAT $\leqslant_p$ Mute Prison Problem.

Properties of Reduction

Suppose that ϕ is an instance of 3-SAT and C_1, C_2, ..., C_m are the clauses of ϕ. By construction of 3-SAT in CNF we have $C_i = (z_{i1} \vee z_{i2} \vee z_{i3})$. In the reduction each C_i's boolean value will represent a boolean value for each language, L_i, spoken by some inmate(s), precisely, $L_i = C_i = (z_{i1} \vee z_{i2} \vee z_{i3})$. Each boolean value for L_i has a specific mean:

$$L_i = \begin{cases} 1 & \text{if } L_i \text{ is spoken by at most 1 inmate} \\ 0 & \text{if } L_i \text{ is spoken by at least 1 inmate} \end{cases}$$

Producing L_1, L_2, ..., L_m will take polynomial time since we iterate through each C_i and perform a boolean or operation on each z_i in C_i which takes O(m).

Finally, the mute prison problem requires a matrix T to produce the subset of inmates S. Let T be an m x m matrix, so that no inmates are left without a language. The rows in T will represent inmates and the columns will represent languages such that column i represents L_i. The algortihm that peforms the reduction will iterate through each L_i. If $L_i = 1$ then set T[i, i] = 1, else if $L_i = 0$ then T[1, i] = T[2, i] = ... = T[m, i] = 1. Assigning all inmates to speak L_i, when $L_i = 0$, will guarantee that |S| = 0. Alternatively, ∀ i, if $L_i = 1$ then |S| = m. So that if ϕ is satisfies 3-SAT, then T will satisfy the mute prison problem if we set k = m. Again this process is polynomial as it iterates through m L_i's and assigns at most m inmates the language L_i, so it will run $O(m^2)$.

ϕ of 3-SAT is satisfiable → L and k of mute prison problem is satisfiable

Suppose ϕ of 3-SAT is satisfiable, then each clause C_1, C_2, ..., C_m is satisfied. A set of L_1, ..., L_m is produced such that ∀ L_i, $L_i = 1$. Then we form matrix T of size m x m, such that T resembles the identity matrix as each T[i,i] = 1. Also, k =m, so that when S is assembled all m inmates speak a different language, then $|S| \geqslant k$ is satisfied.

L and k of mute prison problem is satisfiable → ϕ of 3-SAT is satisfiable

Suppose that T and k of the mute prison problem are satisfiable. Also, suppose |S| is at least m=k. Choose only the first m inmates from S, and extract only their rows from T to form a new matrix T'. It will follows that in T' there will be only m columns where there is at most one entry with the value 1. We will attribute these m columns with variables L_1, ..., L_m, such that, $1 \leqslant i \leqslant m$, $L_i = 1$. We then form m clauses of a 3-SAT CNF, call them C_i, ..., C_m. Each C_i relates to L_i, so that the boolean value of $C_i = (z_{i1} \vee z_{i2} \vee z_{i3}) = 1$. Thus set any one of the z_{i1}, z_{i2}, or z_{i3} to 1. It follows that all $C_i = 1$, thus $\phi = (C_1 \wedge C_2 \wedge ... \wedge C_m)$ Is satisfiable.

So, ϕ of 3-SAT is satisfiable ⇔ L and k of mute prison problem is satisfiable . Also, because the reduction was shown to be polynomial it is proven that the mute prison problem is NP-hard. ■

By the proofs 1. and 2. it follows that the mute prison problem is NP-complete. ■

Q2. The Nonsense Prerequisites

Claim: The nonsense prerequisites problem is NP-complete.

Proof:

1. Show the nonsense prerequisites problem is NP.

2. Show the nonsense prerequisites problem is NP-hard.

1. Suppose we know G(V, E) and k and we are given E' as a certificate. We verify the certificate with the following algorithm:

```
E" = E - E';
Produce function w, such that ∀ (u, v) ∈ E", w(u, v) = -1;
Produce new G'(V, E", w);
for v in V do
    Perform Bellman-Ford(G', w, v);
    for each edge (u, v) ∈ G'.E" do
        if v.d > u.d + w(u, v) then
            There is a cycle and the certificate is not satisfiable.
        end
    end
end
```

If there is a cycle in G'(V, E") then setting each edge in G' to a weight -1 will produce a negative edge cycle which, after relaxations, we can identify easily. Given that G(V, E") may or may not be connected, to locate a cycle in the graph we must perform the relaxation with Bellman-Ford |V| times. Bellman-Ford runs at O(VE), it is executed |V| times in the verifier, thus we have $O(V^2E)$ for our algorithm. Since |V| = n, and $|E| = O(n^2)$, the verifier runs $O(n^4)$. So the verifier is polynomial and then the nonsense prerequisites problem is NP. ■

2.

Q3. T-rex Christmas

Q4. Vertex Cover