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 L 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, $\forall$ i, $L_i = 1$ 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 $O(m^2)$.

Q2. The Nonsense Prerequisites

Q3. T-rex Christmas

Q4. Vertex Cover