CSC373 Winter 2015 Assignment # 2

Name: Weidong An
Student Number: 1000385095

Question 2

Proof: To prove that GoldDigger is NP-complete, we need to show,

1. GoldDigger $\in NP$
2. GoldDigger is NP-hard.

(1) Show that GoldDigger $\in NP$.
Consider the verifier algorithm. The certificates are all the paths $C = [j_1, ..., j_\ell]$ for $\ell \leq m$.

GoldDiggerVerify(h, g, H, G, C)
    $hardness = 0$
    $gold = 0$
    for $k = 1, ..., \ell$
        if not $1 \leq j_k \leq n$
            return FALSE
        if $k \geq 2$ **and not** $j_{k-1} - 1 \leq j_k \leq j_{k-1} + 1$
            return FALSE
        $hardness = hardness + H[k, j_k]$
        $gold = gold + G[k, j_k]$
    if $hardness > h$ **or** $gold < g$
        return FALSE
    return TRUE

GoldDiggerVerify(h, g, H, G, C) returns TRUE for some C if and only if there is a desired drill path.
This algorithm runs in worst case polynomial time because the for loop iterates at most m times and it takes at most polynomial time to implement each iteration. Every line not in the for loop runs in polynomial time clearly.
Hence, GoldDigger $\in NP$.

(2) Show that GoldDigger is NP-hard.
Claim: SubsetSum $\leq_p$ GoldDigger.

The definition of SubsetSum decision problem:
Input: A finite set of positive integers S and a positive integer target t.
Output: Is there some subset S' of S whose sum is exactly t?

Consider the reduction function taking the input of SubsetSum to the input of GoldDigger:

SUBSETSUMTOGOLDDIGGER(S, t)

```
Let H and G be [|S| × 2] arrays
i = 1
for each k ∈ S
    H[i, 1] = k
    G[i, 1] = k
    H[i, 2] = 0
    G[i, 2] = 0
    i = i + 1
h = t
g = t
return (h, g, H, G)
```

s_1, s_1	0,0
s_2, s_2	0,0
$\vdots$	$\vdots$
s_n, s_n	0,0

Figure 1: On input $S = \{s_1, s_2, ..., s_n\}$, output H and G as above.

Runtime: The reduction function runs in worst case $O(n)$ ($n = |S|$) which is in polynomial time.

Correctness

Let $S = \{s_1, s_2, ..., s_n\}$.

Suppose there is a subset S' of S whose sum is exactly t. Define

$$j_k = \begin{cases} 1, & s_k \in S' \\ 2, & s_k \notin S' \end{cases}$$

for $k \in \{1, 2, ..., n\}$. Then $j_1, j_2, ..., j_n$ is a path such that

- $1 \leq j_k \leq 2$ for $k = 1, 2, \ldots, n$ (each coordinate on the path is valid);
- $j_{k-1} - 1 \leq j_k \leq j_{k-1} + 1$ for $k = 2, \ldots, n$ (each block is underneath the one just above, either directly or diagonally) since there are only two columns in both H and G;
- $H[1, j_1] + H[2, j_2] + \cdots + H[n, j_n] = \sum_{i \in S'} i = t \leq h = t$
- $G[1, j_1] + G[2, j_2] + \cdots + G[n, j_n] = \sum_{i \in S'} i = t \geq g = t$

Hence, $j_1, j_2, ..., j_n$ is a desired drill path for input (h, g, H, G).

In the other direction, suppose there is a drill path $j_1, j_2, ..., j_\ell$ for $\ell \leq n$. Since there are only two columns in both H and G, either $j_k = 1$ or $j_k = 2$ for $k = 1, \ldots, \ell$. Construct S' as follows:

$S' = \emptyset$
for $i = 1, \ldots, \ell$
    if $j_i == 1$
        $S' = S' \cup \{s_i\}$

Note that $\sum_{i \in S'} i = G[1, j_1] + G[2, j_2] + \cdots + G[\ell, j_\ell] = H[1, j_1] + H[2, j_2] + \cdots + H[\ell, j_\ell]$.
Also, $t = g \leq G[1, j_1] + G[2, j_2] + \cdots + G[\ell, j_\ell] = H[1, j_1] + H[2, j_2] + \cdots + H[\ell, j_\ell] \leq h = t$.
Then, $\sum_{i \in S'} i = t$.
Then, S' is a subset of S whose sum is exactly t.

Hence, SubsetSum $\leq_p$ GoldDigger.
It is proved that SubsetSum is NP-hard, so GoldDigger is NP-hard.

By (1) and (2), GoldDigger $\in NP$ and GoldDigger is NP-hard.
Hence, GoldDigger is NP-complete. ■