CSC373 Winter 2015 Problem Set # 7

Name: Weidong An
Student Number: 1000385095
UTOR email: weidong.an@mail.utoronto.ca
March 10, 2015

1. (a) Definition of problem $\overline{\text{FRUGAL}}$:

Input: A set of *ingredients* $G = \{g_1, g_2, \ldots, g_m\}$, a set of *recipes* $R = \{r_1, r_2, \ldots, r_n\}$, where each recipe is a subset of ingredients ($r_i \subseteq G$), and a positive integer M.

Output: Are all subsets of recipes $R' \subseteq R$ with size $|R'| \leq M$ such that all together, the recipes in R' cannot use up exactly the ingredients from G ($\bigcup_{r \in R'} r \neq G$ or $r_1 \cap r_2 \neq \emptyset$ for some $r_1, r_2 \in R'$)?

(b) Verifier for FRUGAL: R' is a subset of R.

FRUGAL-VERIFIER(G, R')

1 **if** $|R'| > M$
2   **return** FALSE
3 **for** each $g \in G$
4   **if** $g \notin \bigcup_{r \in R'} r$
5   **return** FALSE
6 **for** each $r_1 \in R$
7   **for** each $r_2 \in R$
8   **if** $r_1 \neq r_2$ and $r_1 \cap r_2 \neq \emptyset$
9   **return** FALSE
10 **return** TRUE

Correctness

The if-statement in line 1 checks whether $|C| \leq M$. If not, FALSE is returned.
The first for-loop checks whether $\bigcup_{r \in R'} r = G$ by check whether each element in G is in $\bigcup_{r \in R'} r$. If one element in G is not in $\bigcup_{r \in R'} r$, FALSE is return. By the input of the problem $\bigcup_{r \in R'} r \subseteq G$. So $\bigcup_{r \in R'} r = G$ if the loop is not returned in the middle.
The second for-loop checks whether any two sets in R' are disjoint. If there are 2 sets that are not disjoint, FALSE is returned. If this loop reaches the end, any two sets in R' are disjoint.
Hence, the algorithm return TRUE if and only if the three conditions above are satisfied. It is a verifier for FRUGAL because FRUGAL-VERIFIER$(G, R') =$ TRUE for some $R' \subseteq R$ if and only if there is some subset of recipes $R' \subseteq R$ with size $|R'| \leq M$ such that all together, the recipes in R' use up exactly the ingredients from G.

Runtime

Let $|G| = m$ and $|R| = n$. The if-statement in line 1 takes $O(n)$ time to check.
The first for-loop: It takes $O(mn)$ time to get the union in line 4. It takes $O(m)$ time to check whether $g \notin \bigcup_{r \in R'} r$ because $|g \notin \bigcup_{r \in R'} r| \leq m$. Totally, the first for loop runs in $mO(mn) = O(m^2 n)$.
The second for-loop: It takes $O(n^2)$ time to check the if-statement in line 8. So, totally the the second for-loop runs in $n^2 O(n^2) = O(n^4)$.

Hence, the algorithm runs in $O(m^2n+n^4)$ time in the worst case which is in polynomial time.

2. (a) Definition of problem $\overline{\text{SHORTPATHS}}$:

   Input: An undirected graph $G = (V, E)$ and a positive integer k.

   Output: Is there a simple path in G contain more than k edges?

   (b) Verifier for SHORTPATHS: C is a simple path in G.

```
ShortPaths-Verifier(G, C)
if |C| ≤ k
        return TRUE
return FALSE
```

Correctness
Correctness is clear by the definition of the problem.
It is a verifier for SHORTPATHS because SHORTPATHS-VERIFIER$(G, C) =$ TRUE for every simple path C if and only if every simple path in G contain at most k edges.
Runtime
It takes $O(m)$ where $m = |E|$ to check the if-statement. So the algorithm runs in $O(m)$ which is in polynomial time.