Assignment 2

CSCI 411

1.

a. Using bellman-ford algorithm, keep track of all the distances. If distance is further updated, loop through the updated node as the starting point and append the values.

b. function findSetI(A)

```
for each vertex V in A
  distance[V] <- infinite
    previous[V] <- NULL
distance[S] <- 0


While i < sizeA
 checkDistanceUpdated <- -1
  for each vertex V in G
    for each edge (U,V) in G
      tempDistance <- distance[U] + edge_weight(U, V)
      if tempDistance < distance[V]
        distance[V] <- tempDistance
        previous[V] <- U
i++
```

```
If distance not updated:
    Return distance_nodes[]
Else:
    For each x in distance:
        append distance_nodes[]

    For i in A:
        if (distance_nodesSize == 0)
                for j in distance_nodesSize
                        append distance_nodes[]


return distance_nodes[]
```

c. My pseudo code is correct because I am passing all the test cases for the code section.

d. O(VE*n) ~ O(n^3), v — vertices, E - edges, each loop will iterate n times.

2.

a. Kruskal: {C, F}, *{E, D}, {F, H}, *{A, B}, {E, H}, {E, G}, {A, C}, {G, I}
   Prims: {A, B}, {B, E}, {E, D}, {E, H}, {H, F}, {F, C}, {E, G}, {G, I}

b. Kruskal: *{A, D}, *{C, F}, *{C, G}, *{C, D}, {B, D}, {C, E}
   Prims: {A, D}, *{D, B}, {D, C}, *{C, F}, *{F, G}, {C, E}

c. Kruskal: *{B, C}, *{B, D}, {B, E}, {F, H}, *{F, I}, {F, G}, {C, A}
   Prims: {A, C}, {C, B}, *{B, D}, {B, E}

3.

a. Considering $G = (V, E)$, for each vertex v, add two vertices $v(in)$ and $v(out)$ with an edge $v(in) \to v(out)$ with capacity equal to that of v. Replace all the edges with $v(in)$ and $v(out)$ with capacity infinity.
Given, any flow through the original network corresponds to a flow with the same value in new network. It is computable in linear time.

b. Input: n^2

Number of undirected edges: $2n^2 - 2n$

G has $2 + n^2 = O(n^2)$ vertices

G has $m + (4n^2 - 4) + 2(2n^2 - 2n) = m + 8n^2 - 4n - 4 = O(n^2)$ edges

Runtime time $O(n^2)$

Using Ford-Fulkerson $O(E+V)$,

$O((E+V). F)$, F being the maximum bound

$F \le 4n - 4$ and $E + V = O(n^2)$

Hence, the higher bound runtime is $O((n^2). (4n - 4)) \sim O(n^3)$.