Stevens Institute of Technology

Department of Computer Science

CS590: Algorithms

Prof. Iraklis Tsekourakis

Homework Assignment 6

Submitted by: Malav Shastri, CWID:10456244

Question 2.

Solution:

We can use Bellmanford to check negative weight cycle but that is after graph has been fully created. Here in our implemented random_graph algorithm if we have been already provided a method which can check whether the given edge is causing negative weight cycle or not we can put while(True) right before generations of random weights and after generating the random weights and assigning it to edge we will call the given the function if the output of the function will be false i.e. there won't be any negative weight cycle we will break that particular while loop else it well keep running and random weights will keep generated for the given edge [u][v] until it will find right pair of weights.

```
if(m_edge[pred][succ]==INT_MAX && count<m){ //Check whether
randomly generated edges do not exceed given edge value m
     while(True){
          random_generator rg;
          int random_weight;
          rg >> random_weight;
          int heighest = w;
          int lowest = -w;
          random_weight = random_weight%(2*heighest+1)+lowest;
          set_edge(pred,succ,random_weight);    //Call set edge
to set the edge.
          if(!GivenFunction()){
               break;
          }
          count++;
        }
```

Question 2.

Solution: The running time for Bellman_Ford algorithm is O(VE) where as we are taking adjacency matrix O(E) = V*V so effectively running time is O(V^3)