Assignment 4 - Graph

PROGRAM ANALYSIS AND OUTPUTS

Atul Bunkar | 214101011 | M.tech, Sem 1

Functions Description:

solution 1:

- Void dfs(): perform dfs on graph. It always starts with the lowest valued node in the graph.
- void traverse_dfs(int v): recursive function for dfs on given node v
- void mark_edge(): marks edge type(tree,forwad,etc) and print dot code to get graph image.

Solution 2:

- void tarjan(), void tar_jan(int i); : Performs tarjan algorithm based on the following pointers:

- void printComponents(): Prints all SCCs of graph in image g_component.png.

Solution 3:

- void minGraph(): Here I created a new graph gmin and filled it with edges from original graph, such that it fulfills the 3 conditions of question 3.

First , I filled the edges in every component in a cycle so as to get a SCC with minimal edges. Then, checked in original graph for inter SCC edges and added atmost 1 edge between them, thus the resulting graph g_min has minimal edges, has same no of SCC , and has the same component graph.

It creates 3 graph images of graph g_min. These are:

1. gmin_graph.png : The graph g_min. This is graph g with minimal edges.

2. gmin_components.png : Shows the SCCs of g_min. You can check the no of SCC are same as with g_components.

3. gmin_component_graph.png : This shows connections between all SCCs. You can check the DAGs of g_Component_Graph and gmin_component_graph are same. The SCC labels(number) will be changed as its not upto me to keep same label for g and g_min.

- bool path_present(int s, int d): checks if there is path from s to d or not. Based on this, all edges are filled in gmin.

Solution 4:

The algo used for this is:

Algorithm 7 IS-SEMI-CONNECTED(G)

```
Compute the component graph of G, call it G'
Perform a topological sort on G' to get the ordering of its vertices
v_1, v_2, ..., v_k.
for i=1..k-1 do
    if there is no edge from v_i to v_{i+1} then
        return FALSE
    end if
end for
return TRUE
```

Working:

I created a new graph g4 which contains all SCCs of graph ‘g’ as nodes and edges between them if any. This forms the component graph of g. This is done in O(V+E).

V-> nodes, E->edges.

Next, topological sort is done on graph g4 to get the ordering of SCC. T.C : O(x+e).

X = no of SCC << no of nodes V. At worst case it can be = no. Of nodes V. Then T.C = O(V+E)

Finally, we check if there is any edge between two adjacent nodes in topological sort or not.

If not, then the graph is not semi-connected, else it is semi-connected. T.C -> O(V) worst case.

So, Overall T.C = O(V+E)

Functions used:

- bool soln4(): just calls the below function.
- bool isSemiCon(): creates component graph.
- void topologicalSort(): topological sort on component graph
- void dfs_topological(int v): helper for above
- bool checklinearorder(): checks if a linear path present in component graph or not.
- void printComponentGraph(): prints component graph dot code.

Solution 5:

void shortestPath(int src,int dest) : Finds shortest path from src to dest , using Dijkstra with min priority queue in O(ElogV) time and prints it.

Important Property:

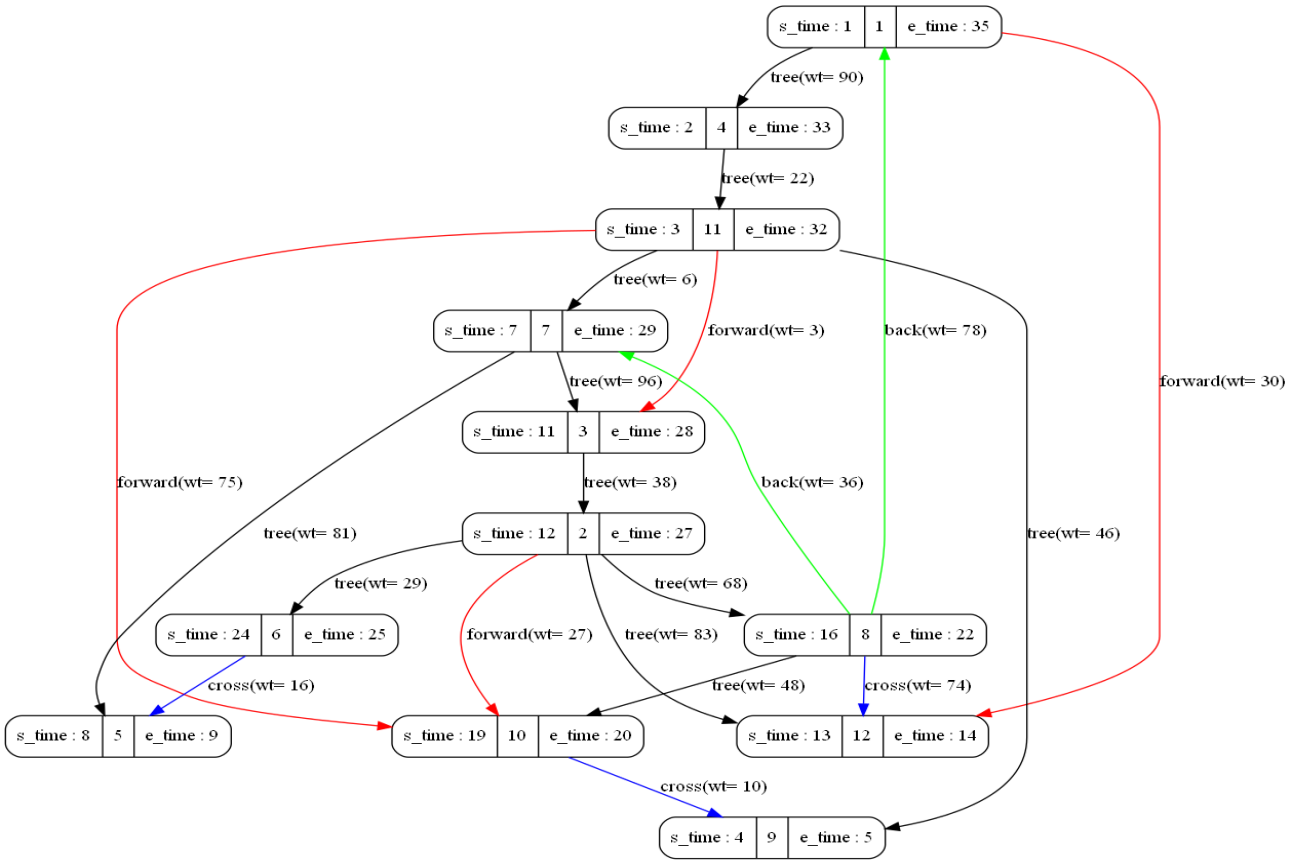
a. Whenever distance of a vertex is reduced, we add one more instance of vertex in priority_queue. Even if there are multiple instances, we only consider the instance with minimum distance and ignore other instances.
b. The time complexity remains O(ELogV)) as there will be at most O(E) vertices in priority queue and O(Log E) is same as O(Log V)

Graph images generated for my given “input.txt” :

From Soln1:

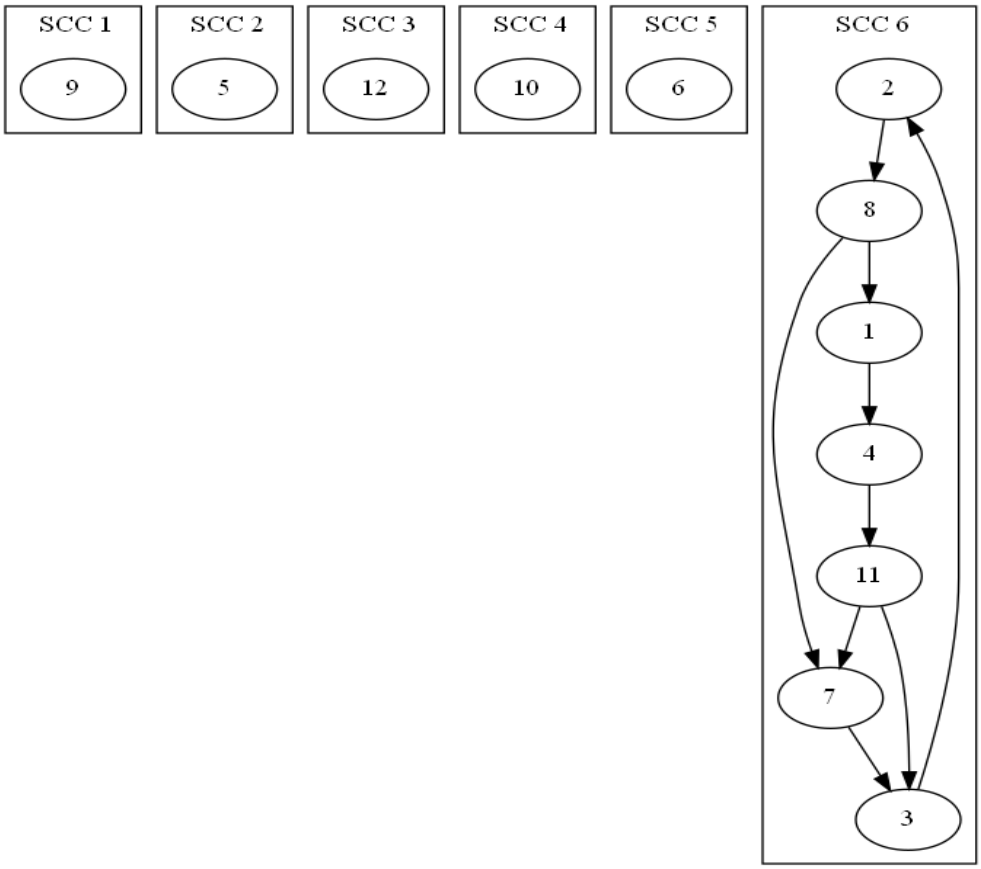
g_Graph.png - DFS graph of graph g.

Structure is : start_time | Node | end_time



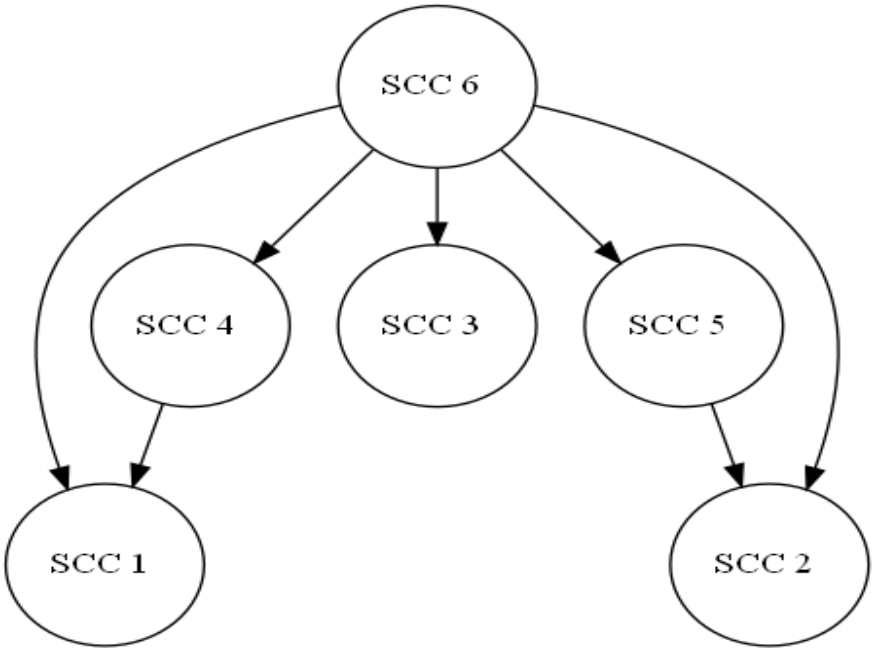
From Soln2:

g_components.png - SCComponents of g.



From soln 4:

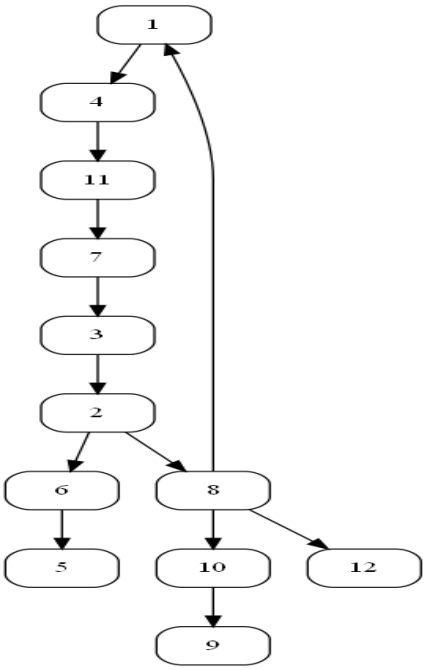
g_Component_Graph – of graph g.



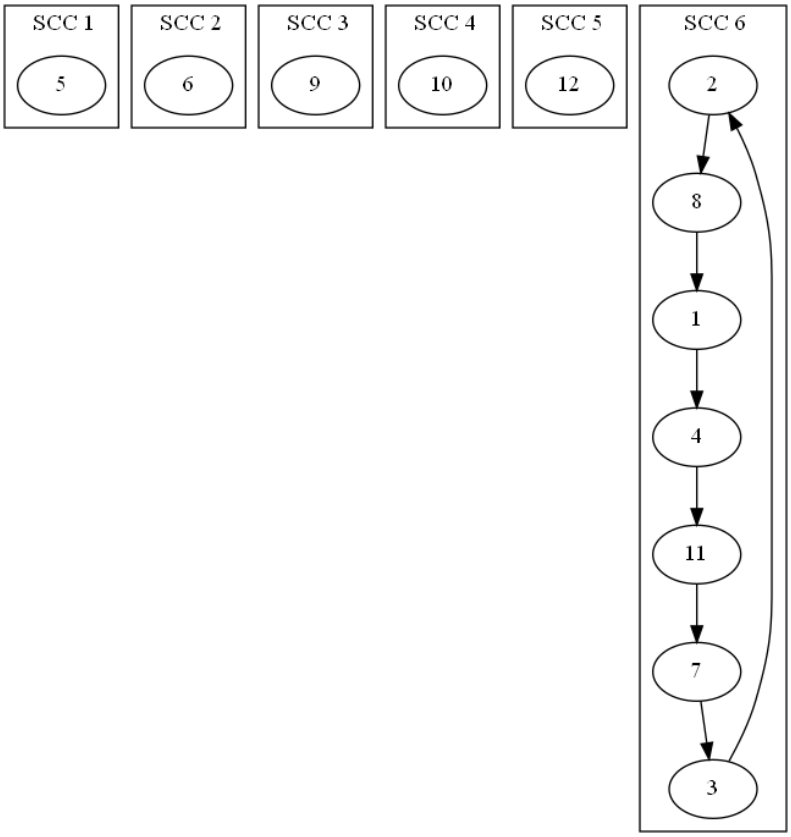
From soln 3:

3 images of graph g_min. As exlained above, it can be used to compare with above 3 images of graph g.

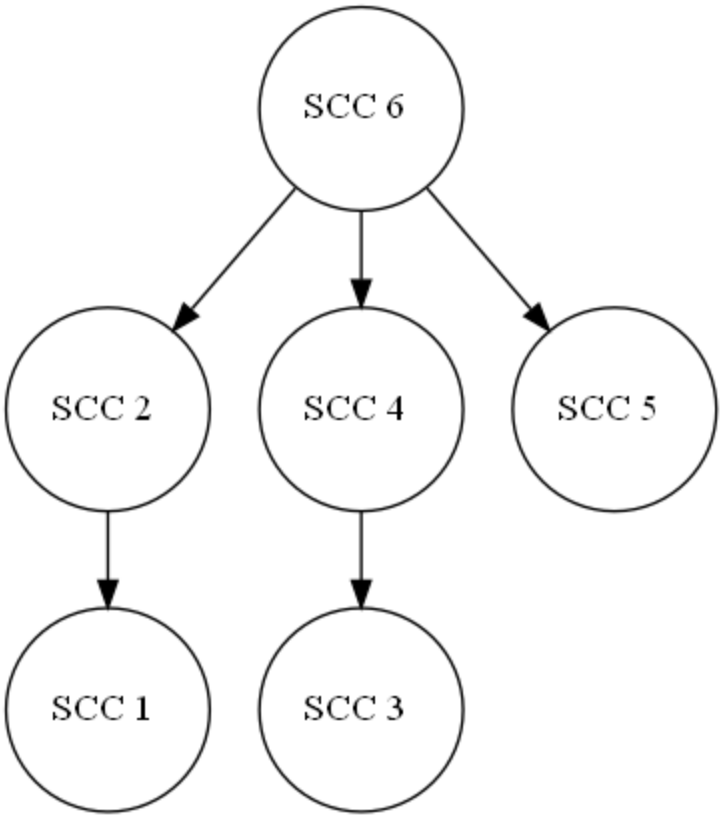
gmin_Graph:



gmin_components:



gmin_Component_Graph:



Note : The Component_graph of g and g_min will be same but component label(numbering) will be different as it is not upto me to fix the labels of SCC.

You can check the difference in component label from g_components.png and gmin_components.png

END OF ANALYSIS !!!