ESO207 Programming Assignment 3

Devansh Kumar Jha(200318) and Divyansh Gupta(200351)

2021–11-8

1 Solution to Part A

1.1 Data Structure Used

A array of HEAD pointers is used where every HEAD represents a vertex of the graph and it points to a linked list storing the vertices connected to that particular vertex. This kind of graph representation is called as **Adjacency List Representation**. At the core the implementation is run fully through structures and pointers only.

1.2 Strategy for solving the problem

- It is pre-assumed that the graph G is bipartite and the first un-visited vertex is assumed to be in the partition V_1 of graph.
- All the un-visited vertices of the graph G which are connected to the current vertex are made a part of the partition other than the current vertex.
- In the above step if a vertex which is already assigned a partition is encountered then it is checked for consistency with the current changes. If inconsistent then our initial assumption is wrong and the graph is not bipartite.
- If the situation in step 3 does not arise then we go back to step 2 and perform it for all the vertices which were alloted there partition by the previous execution of that step.

By this strategy the non-connected vertices or components of the graph are automatically assigned to the partition V_1 and then continued however these vertices are actually floating and they can be kept any of the two partitions giving rise to multiple partition possibilities.

1.3 Main Functions used and Variables Defined

1.3.1 Functions Used

- **Bipartite(G)**
  This returns the graphs V_1 and V_2 which are the partition of the set V of vertices of $G(V, E)$ in case G is bipartite otherwise returns a NULL.

1.3.2 Variables Defined

- hello

1.4 Pseudocodes

1.5 Runtime Analysis

2 Solution to Part B

Answer to this part is given assuming the graph $G(V, E)$ is **Bipartite**.

2.1 Answer

If the graph G is connected then the partitions created for this graph will be unique however for a un-connected graph it will not be unique.

2.2 Explananation

According to the flow of algorithm it is easy to see that when a graph is connected, as soon as we assume one of its veritces to be the part of one of the partitions all other vertices will have to join a partition accordingly and the partition each vertex should be kept into will be well defined. Even if we change the assumption for the first node it will result only on the shuffling of the partitions and not a change in there sets. However, in case of un-connected graph, as soon as we encounter a vertex whose partition cannot be determined from the previous vertices visited there generates a possibility for putting this to anyone of the partition and thus the final sets formed would not be unique.