Name : Kiran K. Patil Course :- parallel computing Lab.

ID : 211070904 sem :- 06

Assignment 02

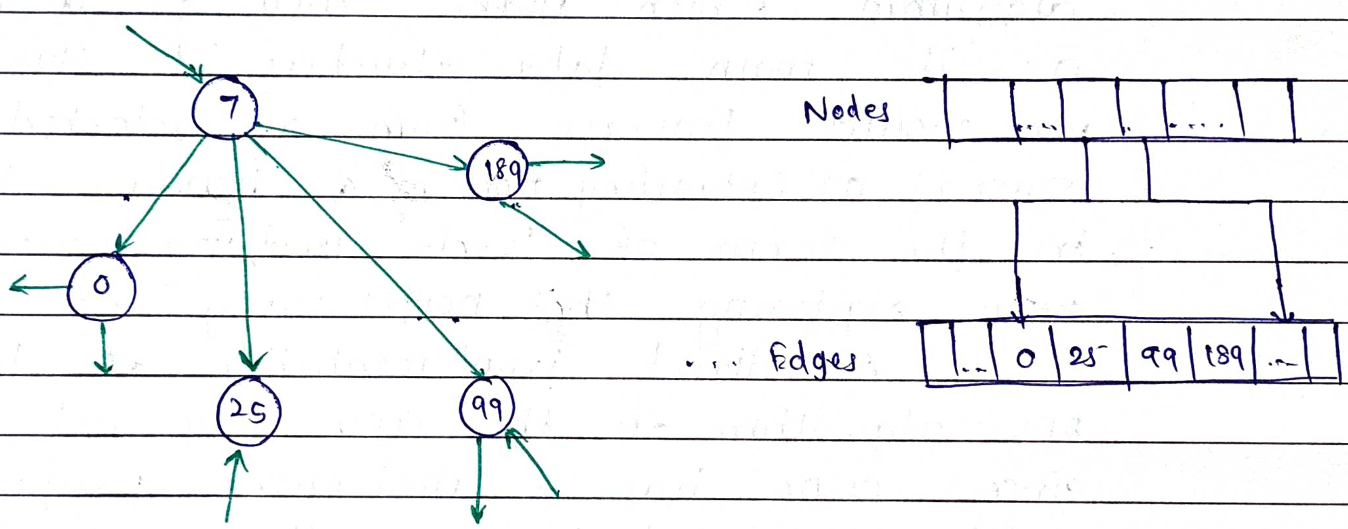
Aim: Implementation of parallel search algorithm BFS using CUDA

Theory : BFS is a well-known graph traversal algorithm which makes used of a queue as its main data structure. In this approach we should traverse from a selected node (source) or (starting node) & traverse the graph in the form of levels implying. We need to keep exploring the neighbouring nodes.

A direct implementation of traditional BFS algorithm on the GPU may not be possible since CUDA has a ~~a~~restrictive programming model and is tricky as well.

Instead of the traditional implementation of the BFS algorithm using a queue and adjecency matrix / List as a graph representation, we have implemented variation of a variation of the BFS algorithm presenting a different data structure to represent the graphs, which is a compact adjecency List form.

In this structure vertices of the graph are present in an array Va. Another array Ea of adjecency lists stores the edge vertices for all vertices in a graph. such that each entry in the vertex array Va refers to ~~storir~~ sorting index of its adjencency neighbouring vertices in Ea (Each entry of Ea refers to vertex in the vertex array va)



In the example the nodes adjacent to 7 are obtained from E[V(7)] to E[v[8]-1]

- Algorithm :

```
cuda_BFS ( Graph G(V,E) , source vertex s)
    create vertex array Va from all vertices and
    edge array Ea form all edges in G(V,E)

    Create frontire arrays Fa, visited array Xa &
    cast array Ca size af va.

    Initialize Fa, Xa, to false and Ca to ∞

    fa[s] ← true , ca[s] ← 0.

    while Fa not Empty do :
        for each vertex v in parallel do
            Invoke cuda_BFS_Kernal (Va,Ea,
                Fa,xa,ca) on the grid.

        end for
    end while.
```

Pseudocode :-

```
Cuda_BFS_Kernal (va, Ea, fa, Xa, ca)
    tid ← getThread ID
    if fa [tid] then
        fa [tid] ← false, xa [tid] ← true

        for all neighbours nid of tid do:
            if Not Xa[nid] then
                ca[nid] ← ca[nid] + 1
                Fa[nid] ← true
            end if
        end for

    end if
```

Conclusion :- Thus we have implemented parallel search algorithm BFS using CUDA.