Implementation Breadth first Search Algorithm

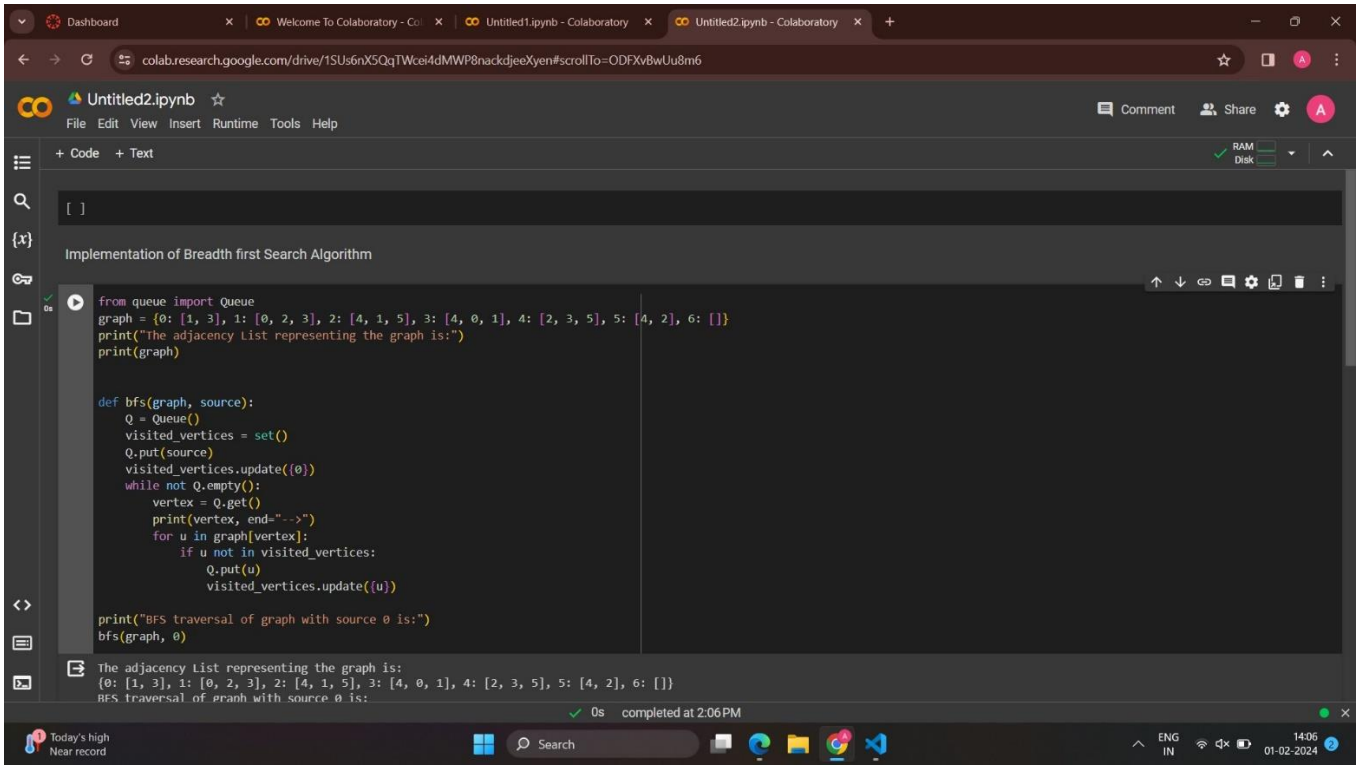
```
from queue import Queue


graph = {0: [1, 3], 1: [0, 2, 3], 2: [4, 1, 5], 3: [4, 0, 1], 4: [2, 3, 5], 5: [4,
2], 6: []}
print("The adjacency List representing the graph is:")
print(graph)


def bfs(graph, source):
    Q = Queue()
    visited_vertices = set()
    Q.put(source)
    visited_vertices.update({0})
    while not Q.empty():
        vertex = Q.get()
        print(vertex, end="-->")
        for u in graph[vertex]:
            if u not in visited_vertices:
                Q.put(u)
                visited_vertices.update({u})


print("BFS traversal of graph with source 0 is:")
bfs(graph, 0)
```

output:-


Dashboard
Welcome To Colaboratory - Col
Untitled1.ipynb - Colaboratory
Untitled2.ipynb - Colaboratory
colab.research.google.com/drive/1SUs6nX5QqTWcei4dMWP8nackdjeeXyen#scrollTo=ODFXvBwUu8m6
Untitled2.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
+ Code + Text
RAM
Disk
[ ]
Implementation of Breadth first Search Algorithm
from queue import Queue
graph = {0: [1, 3], 1: [0, 2, 3], 2: [4, 1, 5], 3: [4, 0, 1], 4: [2, 3, 5], 5: [4, 2], 6: []}
print("The adjacency List representing the graph is:")
print(graph)
def bfs(graph, source):
Q = Queue()
visited_vertices = set()
Q.put(source)
visited_vertices.update({0})
while not Q.empty():
vertex = Q.get()
print(vertex, end="-->")
for u in graph[vertex]:
if u not in visited_vertices:
Q.put(u)
visited_vertices.update({u})
print("BFS traversal of graph with source 0 is:")
bfs(graph, 0)
The adjacency List representing the graph is:
{0: [1, 3], 1: [0, 2, 3], 2: [4, 1, 5], 3: [4, 0, 1], 4: [2, 3, 5], 5: [4, 2], 6: []}
BFS traversal of graph with source 0 is:
0s completed at 2:06 PM
Today's high
Near record
Search
ENG
IN
14:06
01-02-2024

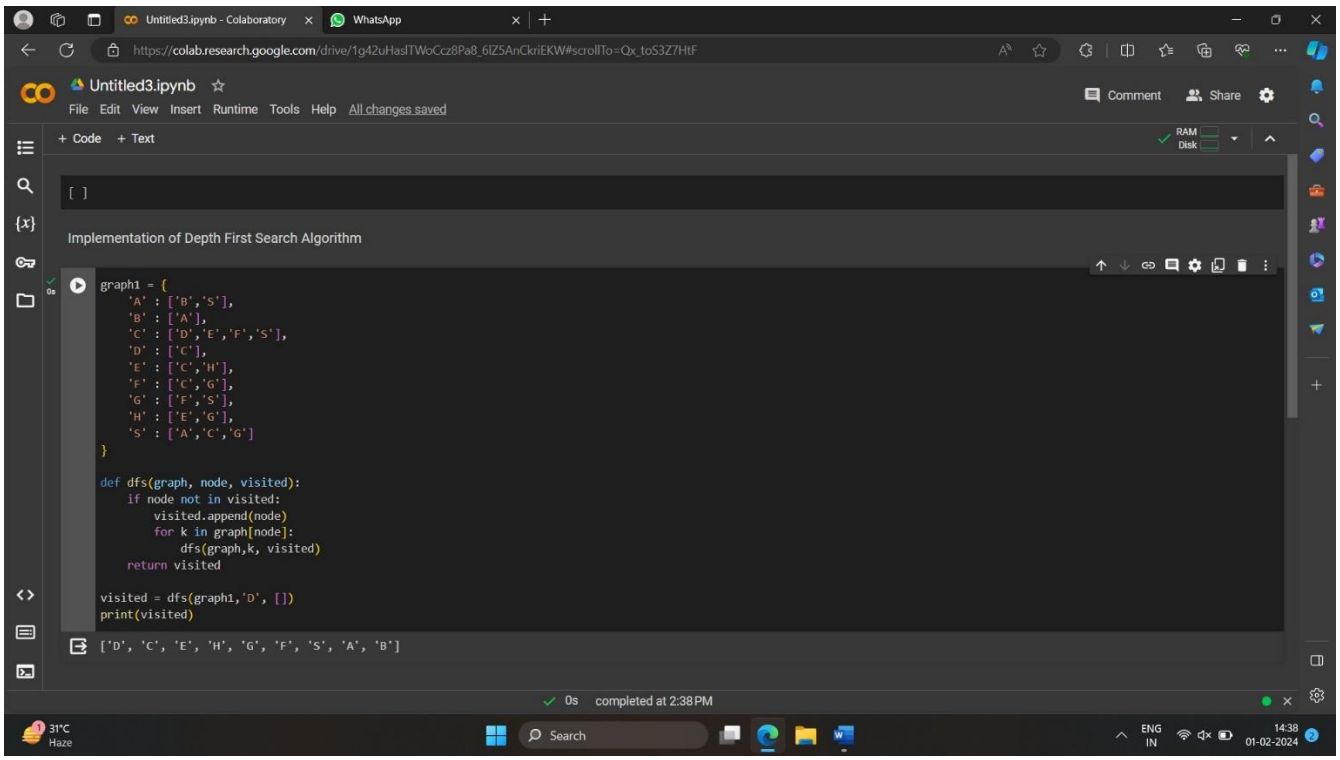
Implementation of Depth First Search Algorithm

```
graph1 = {
   'A' : ['B','S'],
   'B' : ['A'],
   'C' : ['D','E','F','S'],
   'D' : ['C'],
   'E' : ['C','H'],
   'F' : ['C','G'],
   'G' : ['F','S'],
   'H' : ['E','G'],
   'S' : ['A','C','G']
}

def dfs(graph, node, visited):
   if node not in visited:
      visited.append(node)
      for k in graph[node]:
         dfs(graph,k, visited)
   return visited

visited = dfs(graph1,'D', [])
print(visited)
```

output:-


Untitled3.ipynb - Colaboratory
WhatsApp
https://colab.research.google.com/drive/1g42uHaslTWoCcz8Pa8_6IZ5AnCkriEKW#scrollTo=Qx_toS3Z7HtF
Untitled3.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
[ ]
Implementation of Depth First Search Algorithm
graph1 = {
'A' : ['B','S'],
'B' : ['A'],
'C' : ['D','E','F','S'],
'D' : ['C'],
'E' : ['C','H'],
'F' : ['C','G'],
'G' : ['F','S'],
'H' : ['E','G'],
'S' : ['A','C','G']
}
def dfs(graph, node, visited):
if node not in visited:
visited.append(node)
for k in graph[node]:
dfs(graph,k, visited)
return visited
visited = dfs(graph1,'D', [])
print(visited)
['D', 'C', 'E', 'H', 'G', 'F', 'S', 'A', 'B']
0s completed at 2:38 PM
31°C
Haze
Search
ENG
IN
14:38
01-02-2024