

Sprawdzanie czy graf jest dwudzielny BFS

Algorytm : wykonujemy BFS dając obecnemu wierzchołkowi inny kolor niż jego rodzicowi gdy natrafimy na wierzchołek który był już pokolorowany, a teraz musielibyśmy zmienić kolor to oznacza, że graf nie jest dwudzielny.

Złożoność :

Taka jak złożoność algorytmu BFS

Dla reprezentacji listowej $E(V+E)$

Dla reprezentacji macierzowej $E(V^2)$

```
# adjacency list graph representation
from queue import Queue
def BipartiteL(G):
    n = len(G)
    Que=Queue()
    visited=[False for _ in range(n)]
    colors=[-1 for _ in range(n)]
    visited[0]=True
    colors[0]=1
    Que.put(0)

    while Que.qsize() != 0:
        u = Que.get()
        for v in G[u]:
            if colors[v] == -1:
                colors[v] = (colors[u] % 2) + 1
            elif colors[v] == colors[u]:
                return False
            if not visited[v]:
                Que.put(v)
                visited[u]=True

    #check if the graph is complete
    for i in range(len(visited)):
        if not visited[i]:
            return False

    return True

# matrix graph representation
from queue import Queue
def BipartiteM(G):
    n = len(G)
    Que=Queue()
```

```

visited=[False for _ in range(n)]
color=[-1 for _ in range(n)]
visited[0]=True
color[0]=1
Que.put(0)

while Que.qsize():
    u = Que.get()
    for i in range(n):
        if G[u][i]:
            if color[i] == -1:
                color[i] = (color[u] % 2) + 1
            elif color[i] == color[u]:
                return False
            if not visited[i]:
                Que.put(i)
                visited[i]=True

#check if the graph is complete
for i in range(len(visited)):
    if not visited[i]:
        return False

return True

GL = [[4,5],
      [5,6],
      [6,7],
      [7],
      [0],
      [0,1],
      [1,2],
      [2,3]]

GM = [[0,0,0,0,1,1,0,0],
      [0,0,0,0,0,1,1,0],
      [0,0,0,0,0,0,1,1],
      [0,0,0,0,0,0,0,1],
      [1,0,0,0,0,0,0,0],
      [1,1,0,0,0,0,0,0],
      [0,1,1,0,0,0,0,0],
      [0,0,1,1,0,0,0,0]]

print(BipartiteM(GM))

```

