

## AI (2180703)

### Tutorial 2

**Name : Vivek Visavadiya**

**Enrollment No. : 170200107124**

**Division/Batch : F/F3**

Que. Write a program to implement BFS ( for 8 puzzle problem or water jug Problem or any AI search problem.)

**Program(practical2.py):**

```
from collections import deque
def BFS (a, b, target):
    m = { }

    isSolvable = False
    path = [ ]

    q = deque()
    q.append((0, 0))

    while (len(q) > 0):
        u = q.popleft()

        if ((u[0], u[1]) in m):
            continue

        if((u[0] > a or u[1] > b or
            u[0] < 0 or u[1] < 0)):
            continue

        path.append([u[0], u[1]])
        m[ (u[0], u[1]) ] = 1

        if (u[0] == target or u[1] == target):
```

```

isSolvable = True

if (u[0] == target):
    if(u[1] != 0):
        path.append([u[0], 0])
    else:
        if(u[0] != 0):
            path.append([0, u[1]])

sz = len(path)
for i in range(sz):
    print("(" + path[i][0], ",", path[i][1], ")")
    break

q.append([u[0], b])
q.append([a, u[1]])

for ap in range (max(a, b) + 1):

    c = u[0] + ap
    d = u[1] - ap

    if (c == a or (d == 0 and d >= 0)):
        q.append([c, d])

    c = u[0] - ap
    d = u[1] + ap

    if ( (c == 0 and c >= 0) or d == b):
        q.append([c, d])

q.append([a, 0])
q.append([0, b])

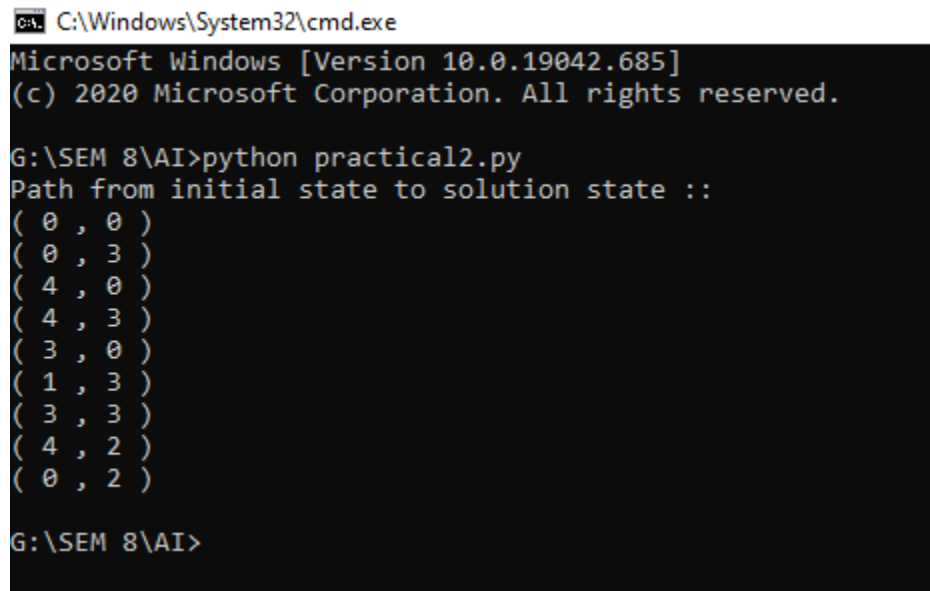
if(not isSolvable):
    print("No solution")
if __name__ == '__main__':

```

```
print("Rule : x > y and x > target")
x = int(input("Enter x : "))
y = int(input("Enter y : "))
target = int(input("Enter target : "))
print("path from initial state to solution state ::")

BFS(x, y, target)
```

## OUTPUT:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19042.685]
(c) 2020 Microsoft Corporation. All rights reserved.

G:\SEM 8\AI>python practical2.py
Path from initial state to solution state ::
( 0 , 0 )
( 0 , 3 )
( 4 , 0 )
( 4 , 3 )
( 3 , 0 )
( 1 , 3 )
( 3 , 3 )
( 4 , 2 )
( 0 , 2 )

G:\SEM 8\AI>
```