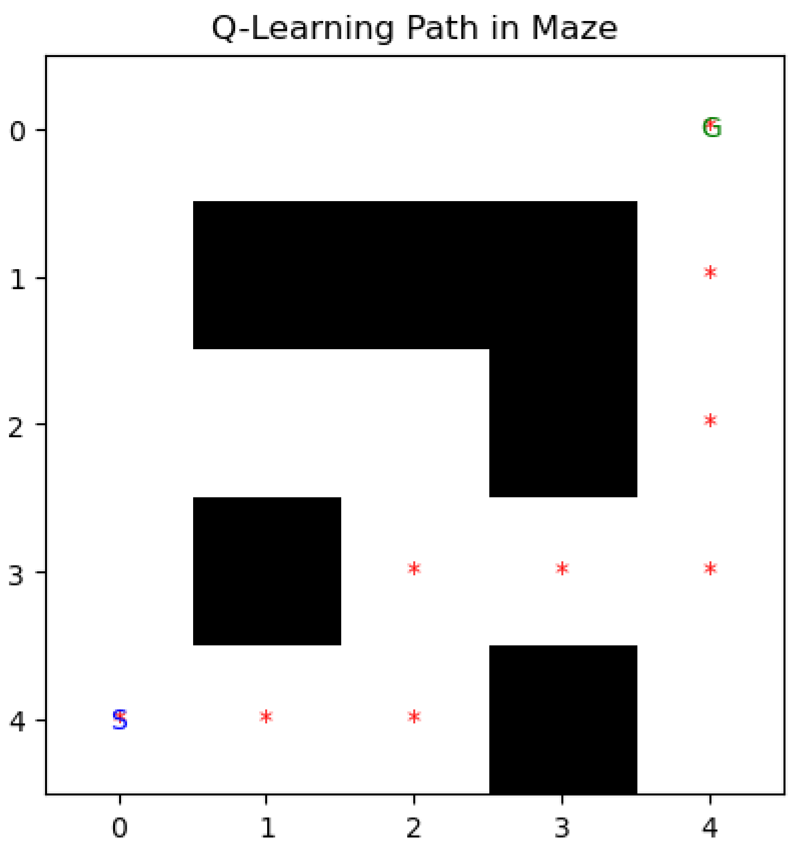
In [ ]:

```python
# Vidhi Rane C-11
```

In [1]:

```python
import numpy as np, random, matplotlib.pyplot as plt

# Maze: 0=free, 1=wall
grid = np.array([
    [0,0,0,0,0],
    [0,1,1,1,0],
    [0,0,0,1,0],
    [0,1,0,0,0],
    [0,0,0,1,0]
])
start, goal = (4,0), (0,4)
actions = [(-1,0),(1,0),(0,-1),(0,1)]  # up,down,left,right

# Convert state to index
def idx(s): return s[0]*grid.shape[1]+s[1]

# Q-learning params
Q = np.zeros((grid.size,4))
alpha,gamma,eps = 0.6,0.95,1.0

# Train
for ep in range(2000):
    s = start
    for _ in range(200):
        a = random.randrange(4) if random.random()<eps else np.argmax(Q[idx(s)
        r,c = s[0]+actions[a][0], s[1]+actions[a][1]
        if (0<=r<5 and 0<=c<5 and grid[r,c]==0):
            ns = (r,c)
            reward,done = (10,True) if ns==goal else (-0.1,False)
        else:
            ns,reward,done = s,-0.5,False
        Q[idx(s),a] += alpha*(reward + gamma*np.max(Q[idx(ns)])-Q[idx(s),a])
        s = ns
        if done: break
    eps = max(0.05, eps*0.995)

# Follow greedy policy
path=[start]; s=start
while s!=goal and len(path)<50:
    a=np.argmax(Q[idx(s)])
    s=(s[0]+actions[a][0], s[1]+actions[a][1])
    path.append(s)

# Plot path
plt.imshow(grid,cmap="gray_r")
for (r,c) in path: plt.text(c,r,"*",ha="center",va="center",color="red")
plt.text(start[1],start[0],"S",ha="center",va="center",color="blue")
plt.text(goal[1],goal[0],"G",ha="center",va="center",color="green")
plt.title("Q-Learning Path in Maze")
plt.show()
```



```
In [ ]:
```