

```

In [3]: adj = [
    [0, 1, 1, 0, 0, 0, 0, 0],
    [0, 0, 0, 1, 0, 0, 0, 1],
    [0, 0, 0, 0, 1, 0, 1, 0],
    [0, 0, 1, 0, 0, 0, 1, 0],
    [0, 0, 0, 0, 0, 1, 1, 0],
    [0, 1, 1, 0, 0, 0, 0, 0],
    [0, 0, 0, 1, 0, 0, 0, 1],
    [1, 0, 0, 0, 0, 0, 1, 0]
]

class State:
    def __init__(self, pai, cidade, visitouH, visitouE):
        self.pai = pai
        self.cidade = cidade
        self.visitouH = visitouH or cidade == 4
        self.visitouE = visitouE or cidade == 7

    def initialState():
        return State(None, 0, 0, 0)

    def showState(s):
        cidades = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H']
        print(f'Cidade: {cidades[s.cidade]}, {s.visitouE}, {s.visitouH}')

    def goal(s):
        return s.visitouH != 0 and s.visitouE != 0

    def expand(s):
        ret = []
        cidade = s.cidade
        for i in range(len(adj)):
            if(adj[cidade][i] != 0):
                filho = State(s, i, s.visitouE, s.visitouH)
                ret.append(filho)

        return ret

    def showPath(s):
        if(s == None):
            return
        showPath(s.pai)
        showState(s)

queue = []

def enqueue(s):
    queue.append(s)

def dequeue():
    return queue.pop(0)

s = initialState()
enqueue(s)

while(queue):
    s = dequeue()

    if(goal(s)):
        showPath(s)
        break

    children = expand(s)
    for child in children:
        enqueue(child)

```