

```

In [3]: dists = [
    [0, 15, 30, 0, 0, 0, 0, 0],
    [0, 0, 0, 70, 0, 0, 0, 10],
    [0, 0, 0, 0, 10, 0, 70, 0],
    [0, 0, 20, 0, 0, 0, 20, 0],
    [0, 0, 0, 0, 0, 40, 60, 0],
    [0, 60, 70, 0, 0, 0, 0, 0],
    [0, 0, 0, 10, 0, 0, 0, 15],
    [40, 0, 0, 0, 0, 0, 50, 0]
    ]

class State:
    def __init__(self, parent, cidade, autonomia):
        self.parent = parent
        self.cidade = cidade
        self.autonomia = autonomia

def initialState():
    return State(None,0,80)

def showState(s):
    print("Cidade: ", s.cidade, "Autonomia: ", s.autonomia)

def goal(s):
    return(s.cidade == 5)

def expand(s):
    ret = []
    origem = s.cidade
    for i in range(len(dists[origem])):
        if(dists[origem][i]!=0 and dists[origem][i]<=s.autonomia):
            filho = State(s,i,s.autonomia - dists[origem][i])
            ret.append(filho)
    return ret

def showPath(s):
    if(s==None):
        return
    showPath(s.parent)
    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)

```

| | | | |
|---------|---|------------|----|
| Cidade: | 0 | Autonomia: | 80 |
| Cidade: | 2 | Autonomia: | 50 |
| Cidade: | 4 | Autonomia: | 40 |
| Cidade: | 5 | Autonomia: | 0 |