In [1]:

```
import numpy as np
Q=np.zeros((27,6),dtype=float)
```

In [2]:

```
R=np.zeros((27,6),dtype=int)
R[17,1]=100
R[23,3]=100
```

$$Q'(s_t, a_t) = (1-\nu)Q(s_t, a_t) + \nu[r(s_t, a_t) + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1})]$$

In [3]:

```
v=0.9 # Factor de aprendizaje learning rate
y=0.8 # Factor de descuento discount factor
```

In [4]:

```
# Tabla de transiciones
import pandas as pd
df= pd.read_csv("T.csv",header=None)
T=df.to_numpy()
T
```

Out[4]:

```
array([[ 1,  2, -1, -1, -1, -1],
       [-1,  3,  0,  2, -1, -1],
       [ 4, -1, -1, -1,  0,  1],
       [-1, -1,  5,  6,  1, -1],
       [-1, -1,  2, -1,  8,  7],
       [ 3,  6, -1, -1, -1,  8],
       [ 9, -1, -1, -1,  5,  3],
       [-1, 10,  8,  4, -1, -1],
       [ 7,  4, -1,  5, -1, -1],
       [-1, -1,  6, -1, 11, 12],
       [-1, -1, 13, 14,  7, -1],
       [12,  9, -1, -1, -1, 15],
       [-1, -1, 11,  9, 16, -1],
       [10, 14, -1, 17, -1, -1],
       [-1, -1, 18, -1, 13, 10],
       [19, 20, -1, 11, -1, -1],
       [-1, 12, 21, 22, -1, -1],
       [23, 26, -1, -1, -1, 13],
       [14, -1, -1, -1, 24, 25],
       [-1, -1, 15, 20, -1, -1],
       [-1, -1, 22, -1, 15, 19],
       [16, 22, -1, 24, -1, -1],
       [20, -1, -1, -1, 21, 16],
       [-1, -1, 17, 26, 25, -1],
       [25, 18, -1, -1, -1, 21],
       [-1, 23, 24, 18, -1, -1],
       [-1, -1, -1, -1, 17, 23]], dtype=int64)
```

In [5]:

```
#Seleccionamos un estado al azar
s=0 #Partimos del estado inicial
entrenar=0
while(entrenar<100000):
    a=np.random.randint(6) # Acción aleatoria al azar número entero en [0,5]
    while T[s,a]==-1:
        a=np.random.randint(6)
    # T[s,a] es una transición posible
    siguiente=T[s,a] # Estado siguiente
    Q[s,a]=(1-v)*Q[s,a]+v*(R[s,a]+y*max(Q[siguiente,]))
    # print (s,"-->",siguiente)
    if siguiente!=26:
        s=siguiente # Estado siguiente
    else:
        s=0
    entrenar+=1
```

In [6]:

```
#Seleccionamos un estado al azar
entrenar=0
while(entrenar<100000):
    s=np.random.randint(26) #estado aleatorio [0,25]
    a=np.random.randint(6) # Acción aleatoria al azar número entero en [0,5]
    while T[s,a]==-1:
        a=np.random.randint(6)
    # T[s,a] es una transición posible
    siguiente=T[s,a] # Estado siguiente
    Q[s,a]=(1-v)*Q[s,a]+v*(R[s,a]+y*max(Q[siguiente,]))
    entrenar+=1
```

In [7]:

```
R
```

Out[7]:

```
array([[  0,   0,   0,   0,   0,   0],
       [  0,   0,   0,   0,   0,   0],
       [  0,   0,   0,   0,   0,   0],
       [  0,   0,   0,   0,   0,   0],
       [  0,   0,   0,   0,   0,   0],
       [  0,   0,   0,   0,   0,   0],
       [  0,   0,   0,   0,   0,   0],
       [  0,   0,   0,   0,   0,   0],
       [  0,   0,   0,   0,   0,   0],
       [  0,   0,   0,   0,   0,   0],
       [  0,   0,   0,   0,   0,   0],
       [  0,   0,   0,   0,   0,   0],
       [  0,   0,   0,   0,   0,   0],
       [  0,   0,   0,   0,   0,   0],
       [  0,   0,   0,   0,   0,   0],
       [  0,   0,   0,   0,   0,   0],
       [  0,   0,   0,   0,   0,   0],
       [  0, 100,   0,   0,   0,   0],
       [  0,   0,   0,   0,   0,   0],
       [  0,   0,   0,   0,   0,   0],
       [  0,   0,   0,   0,   0,   0],
       [  0,   0,   0,   0,   0,   0],
       [  0,   0,   0,   0,   0,   0],
       [  0,   0,   0, 100,   0,   0],
       [  0,   0,   0,   0,   0,   0],
       [  0,   0,   0,   0,   0,   0],
       [  0,   0,   0,   0,   0,   0]])
```

```
In [8]: Q
```

```
Out[8]: array([[ 20.97152,  26.2144 ,   0.     ,   0.     ,   0.     ,   0.     ],
               [  0.     ,  20.97152,  20.97152,  26.2144 ,   0.     ,   0.     ],
               [ 32.768  ,   0.     ,   0.     ,   0.     ,  20.97152,  20.97152],
               [  0.     ,   0.     ,  26.2144 ,  20.97152,  20.97152,   0.     ],
               [  0.     ,   0.     ,  26.2144 ,   0.     ,  32.768  ,  40.96   ],
               [ 20.97152,  20.97152,   0.     ,   0.     ,   0.     ,  32.768  ],
               [ 20.97152,   0.     ,   0.     ,   0.     ,  26.2144 ,  20.97152],
               [  0.     ,  51.2    ,  32.768  ,  32.768  ,   0.     ,   0.     ],
               [ 40.96   ,  32.768  ,   0.     ,  26.2144 ,   0.     ,   0.     ],
               [  0.     ,   0.     ,  20.97152,   0.     ,  20.97152,  26.2144 ],
               [  0.     ,   0.     ,  64.     ,  51.2    ,  40.96   ,   0.     ],
               [ 26.2144 ,  20.97152,   0.     ,   0.     ,   0.     ,  20.97152],
               [  0.     ,   0.     ,  20.97152,  20.97152,  32.768  ,   0.     ],
               [ 51.2    ,  51.2    ,   0.     ,  80.     ,   0.     ,   0.     ],
               [  0.     ,   0.     ,  51.2    ,   0.     ,  64.     ,  51.2    ],
               [ 20.97152,  26.2144 ,   0.     ,  20.97152,   0.     ,   0.     ],
               [  0.     ,  26.2144 ,  40.96   ,  32.768  ,   0.     ,   0.     ],
               [ 80.     , 100.     ,   0.     ,   0.     ,   0.     ,  64.     ],
               [ 51.2    ,   0.     ,   0.     ,   0.     ,  51.2    ,  64.     ],
               [  0.     ,   0.     ,  20.97152,  26.2144 ,   0.     ,   0.     ],
               [  0.     ,   0.     ,  32.768  ,   0.     ,  20.97152,  20.97152],
               [ 32.768  ,  32.768  ,   0.     ,  51.2    ,   0.     ,   0.     ],
               [ 26.2144 ,   0.     ,   0.     ,   0.     ,  40.96   ,  32.768  ],
               [  0.     ,   0.     ,  80.     , 100.     ,  64.     ,   0.     ],
               [ 64.     ,  51.2    ,   0.     ,   0.     ,   0.     ,  40.96   ],
               [  0.     ,  80.     ,  51.2    ,  51.2    ,   0.     ,   0.     ],
               [  0.     ,   0.     ,   0.     ,   0.     ,   0.     ,   0.     ]])
```

In [9]:

```
for t in range(0,26):
    print ("s",t," accion:",np.argmax(Q[t,]))
```

```
s 0  accion: 1
s 1  accion: 3
s 2  accion: 0
s 3  accion: 2
s 4  accion: 5
s 5  accion: 5
s 6  accion: 4
s 7  accion: 1
s 8  accion: 0
s 9  accion: 5
s 10  accion: 2
s 11  accion: 0
s 12  accion: 4
s 13  accion: 3
s 14  accion: 4
s 15  accion: 1
s 16  accion: 2
s 17  accion: 1
s 18  accion: 5
s 19  accion: 3
s 20  accion: 2
s 21  accion: 3
s 22  accion: 4
s 23  accion: 3
s 24  accion: 0
s 25  accion: 1
```

In [ ]: