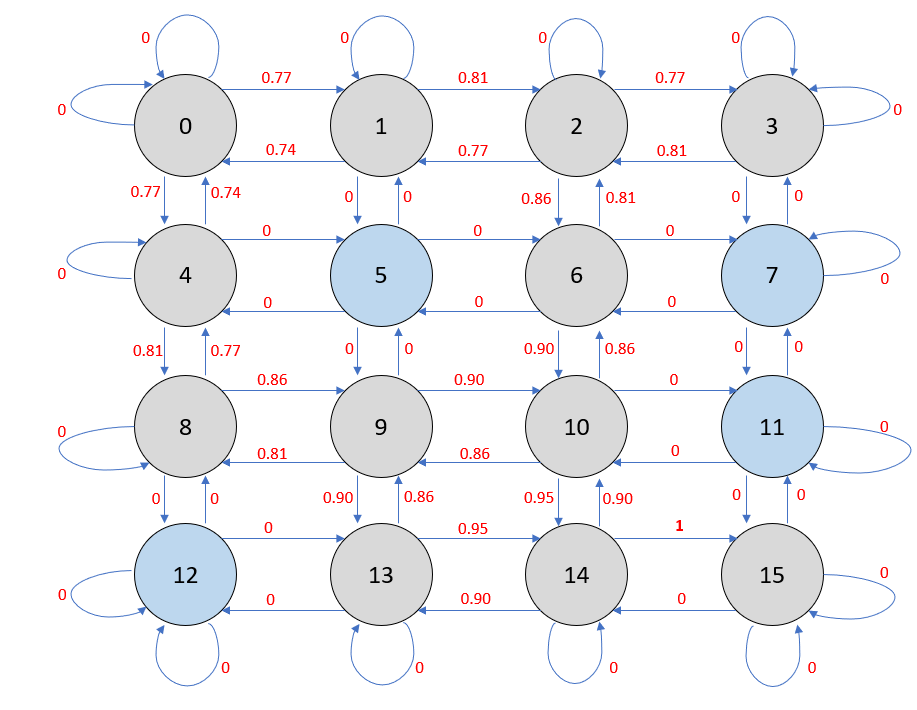
**MEMORIA PRACTICA 4. PARTE 2**

**Artem Vartanov y Daniel Coleto**

***EJERCICIO 2***

***a) Dibujar la máquina de estados final después de haber aplicado el algoritmo QLearning.***



Esta es la máquina de estados final después de 15000 episodios de entrenamiento por refuerzo *Q-Learning*. En el entrenamiento hemos descartado los movimientos en los que el personaje se sale fuera del tablero 4x4, por lo que el peso de transición para esos casos será cero en nuestra máquina.

b) ***Dada la máquina anterior, calcula el camino mínimo para llegar al objetivo. Indica el camino seguido y el número de pasos en los que se llega al objetivo.***

Partiendo desde el estado 0 hasta el estado 15 objetivo, el camino más corto que encontramos es el siguiente:

Sucesión de estados: 0 🡪 4 🡪 8 🡪 9 🡪 13 🡪 14 🡪 15.  
Con esta solución, se llega al estado objetivo en 6 pasos.

c) ***¿Hay varios caminos para llegar al objetivo? Si los hay, indica cuáles son.***

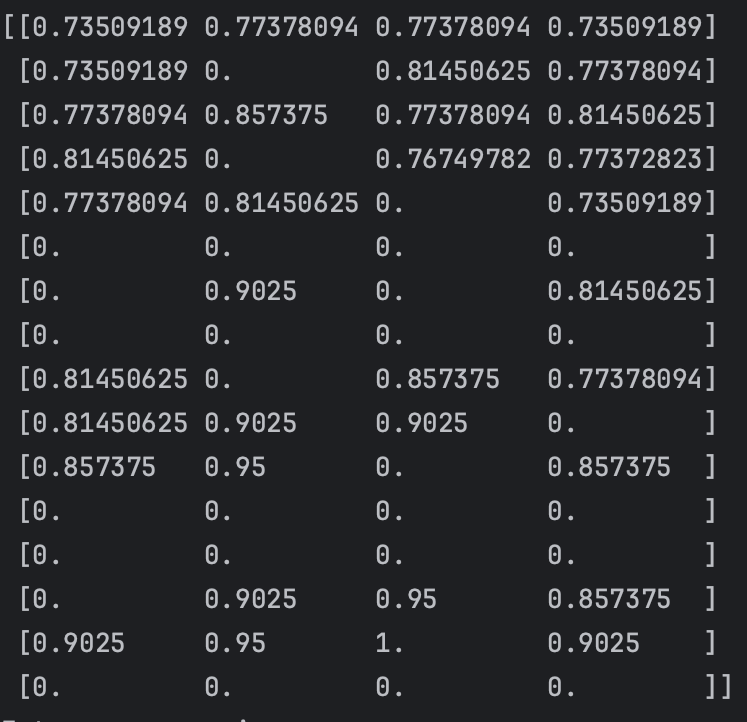
Estos son todos los caminos que hay desde el E0 hasta el E15. Además, marcaremos cuáles son óptimos.

* 0 🡪 4 🡪 8 🡪 9 🡪 13 🡪 14 🡪 15 => **óptimo.** Longitud = 6
* 0 🡪 4 🡪 8 🡪 9 🡪 10 🡪 14 🡪 15 => **óptimo.** Longitud = 6
* 0 🡪 1 🡪 2 🡪 6 🡪 10 🡪 14 🡪 15 => **óptimo.** Longitud = 6
* 0 🡪 1 🡪 2 🡪 6 🡪 10 🡪 9 🡪 13 🡪 14 🡪 15 => **no** **óptimo.** Longitud = 8

***EJERCICIO 3***

A partir de este ejercicio utilizaremos la función *sample()* de la librería para la elección de una acción aleatoria. Esto se hace por el cambio de la distribución de probabilidades entre las acciones posibles cuando en un futuro nuestro parámetro *is\_slippery = True.*

Es por ello que nuestra matriz de transiciones ahora tiene este aspecto:

 Matriz de transiciones para is\_slippery = False.

Vemos que ahora algunos movimientos que nosotros habíamos considerado como ilegales, tienen valores positivos.

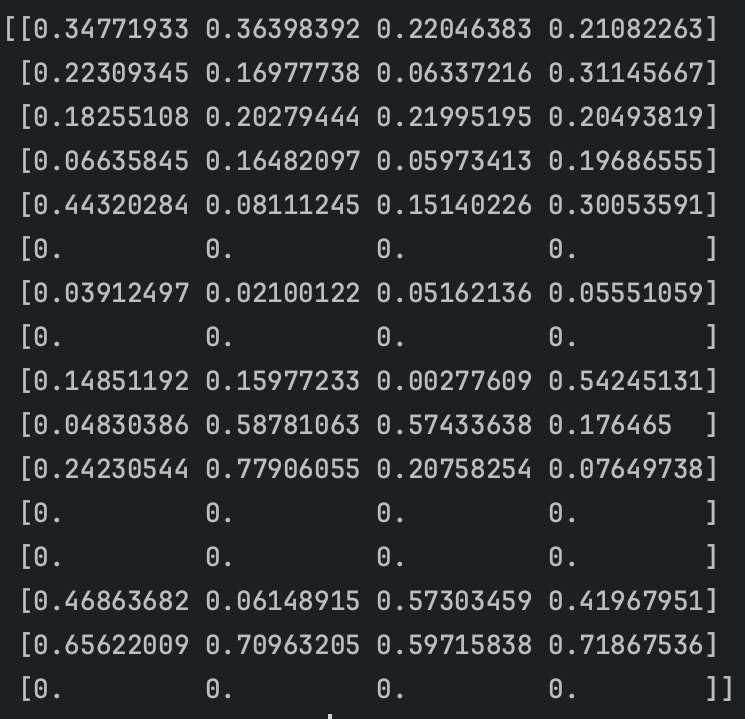
Los caminos mínimos encontrados con este entrenamiento son:

* 0 🡪 4 🡪 8 🡪 9 🡪 13 🡪 14 🡪 15
* 0 🡪 4 🡪 8 🡪 9 🡪 10 🡪 14 🡪 15
* 0 🡪 1 🡪 2 🡪 6 🡪 10 🡪 14 🡪 15

Cada uno de longitud 6 pasos.

***a) Cambiar el parámetro “is\_slippery” a True para utilizar la física del juego.***

1. **¿Se sigue encontrando el camino mínimo en el mismo número de pasos que en el primer caso (is\_slippery = False)? Razona tu respuesta**

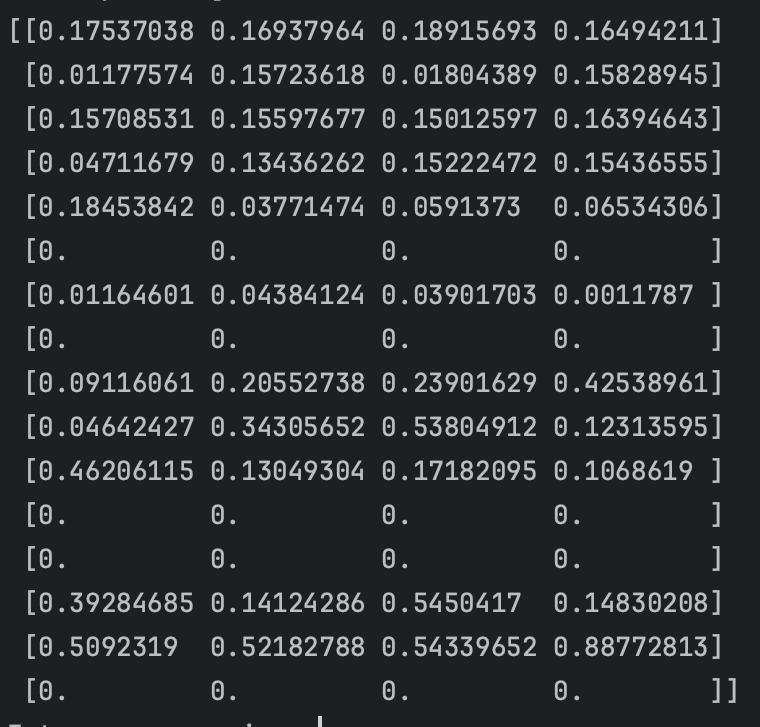


Mirando la matriz Q, al realizar explotación no podemos llegar a la solución, pues nos queda el siguiente camino:

0 🡪 4 🡪 4 ... (infinite loop)

1. **Si aumentamos el número de episodios a un número superior a 15000 ¿mejora la matriz Q?**

No. Sigue entrando en un bucle con la pared. Esta es la matriz obtenida.

0 🡪 1 🡪 1 ... (infinite loop)

***b) Cambiar el tablero a los siguientes tableros y cambiar el parámetro is\_slippery a False.***

*Tablero 1:*SFFH  
HHFH  
FFFH  
FFFG

**a) ¿Qué ha sucedido? Razona por qué la matriz Q no es la esperada.**

La matriz Q sigue siendo igual que al inicializarla, toda llena de 0’s. Esto se debe a que el gnomo nunca llega a coger el regalo por la dificultad del camino solución. Además, por épsilon = 0.3, estamos cogiendo con un 70% de probabilidad una acción por explotación, que teniendo la matriz “vacía”, es siempre izquierda.

**b) ¿Si solo aumentas el número episodios mejora la matriz Q? ¿Qué parámetros tienes que cambiar para obtener una matriz Q mejor?**

Al aumentar el número de episodios a 30000, la matriz Q sigue igual. Cambiamos el parámetro épsilon a 0.7 y nos sale una matriz Q que encuentra la solución óptima.

*Tablero 2:*SHFFFFFFFFHFHFHG

**a) ¿Qué ha sucedido? Razona por qué la matriz Q no es la esperada.**

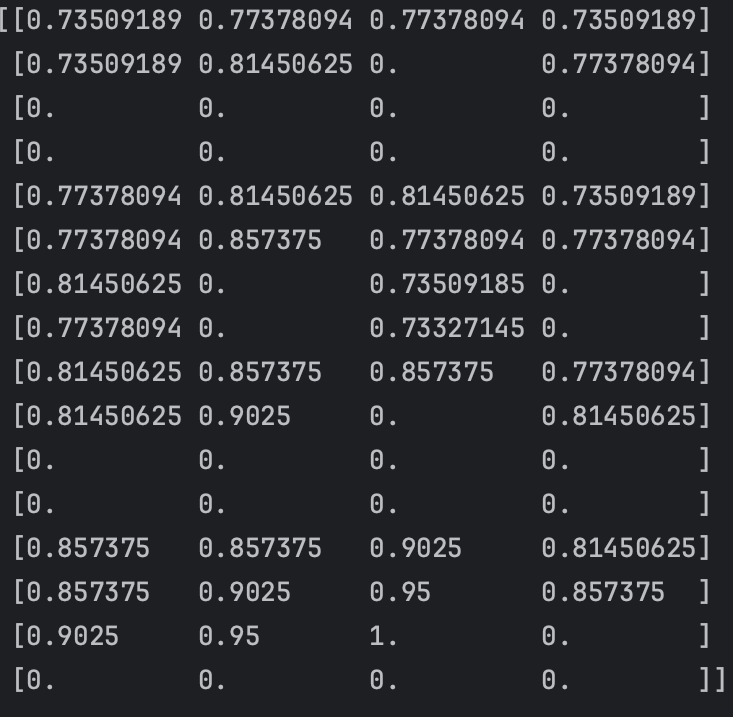
En nuestro caso, usando la matriz Q realizando explotación, llegaremos al camino óptimo a la solución.

**b) ¿Si solo aumentas el número episodios mejora la matriz Q? ¿Qué parámetros tienes que cambiar para obtener una matriz Q mejor?**

Nuestra matriz sigue igual al aumentar el número de episodios a 30000, pues ya ha convergido antes de 15000 episodios.

No necesitamos mejorar la matriz, pues ya nos sale bastante correcto.

***c) Cambiar el tablero a un tablero aleatorio y mantener el parámetro is\_slippery a False.***

Tablero 3 (aleatorio)  
SFHF  
FFFF  
FFHF  
FFFG

El camino mínimo obtenido es:

0 → 4 →8→ 12 → 13→ 14→ **15**

6 pasos