Міністерство освіти і науки України  
НТУУ «КПІ ім. І. Сікорського»

Кафедра автоматизації проектування енергетичних процесів і систем

ЛАБОРАТОРНА РОБОТА № 4  
з дисципліни «МЕТОДОЛОГІЯ РОЗРОБКИ ІНТЕЛЕКТУАЛЬНИХ КОМП’ЮТЕРНИХ ПРОГРАМ»

Тема: «Інтелектуальні агенти. Алгоритм Q-навчання»

ТІ-92 Черноусов Денис

Перевірив д.т.н Мусієнко А. П.

КИЇВ 2021

**Мета роботи:**

Ознайомитися з поняттям інтелектуального агента та одним з

методів його навчання – Q-learning

**Завдання:**

*Завдання 1*

Відповідно до свого варіанта (**6 варіант**) побудувати математичну

модель інтелектуального агента та його зовнішнього середовища:

– записати множину станів інтелектуального агента та цільовий стан;

– зобразити зовнішнє середовище агента у вигляді графа, в якому стани

– вузли, а дії – ребра;

– записати матрицю суміжності побудованого графа.

Проілюструвати роботу алгоритму Q-навчання розрахунками без

програмної реалізації. Для цього показати 2 спроби агента досягти цільової

мети. Перший раз, обираючи маршрут цілком довільним чином (так як пам’ять

агента порожня), другий раз – враховуючи здобуту пам’ять агента. Розрахунки

повинні містити обчислення винагород за дії агента, значення матриці Q та

зображення графа зовнішнього середовища з позначенням шляху агента і

змінених значень ваг ребер.

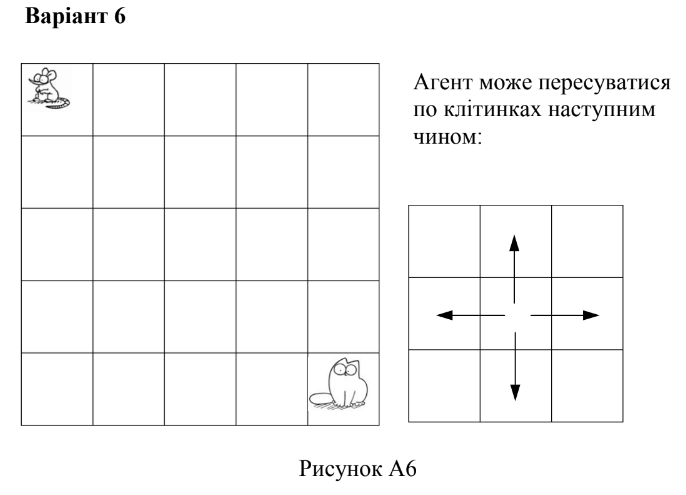
*Завдання 2*

Відповідно до свого варіанта (**6 варіант**) та побудованої в 1

завданні математичної моделі реалізувати програмне забезпечення для

ілюстрації роботи алгоритму Q-навчання. Примітка: на початку програми

ініціалізувати матрицю Q нулями.



**Хід роботи**

**Завдання 1**

Пронумеруємо кожну клітинку

25

13

2

4

3

6

7

12

11

10

5

9

8

1

14

15

16

17

18

19

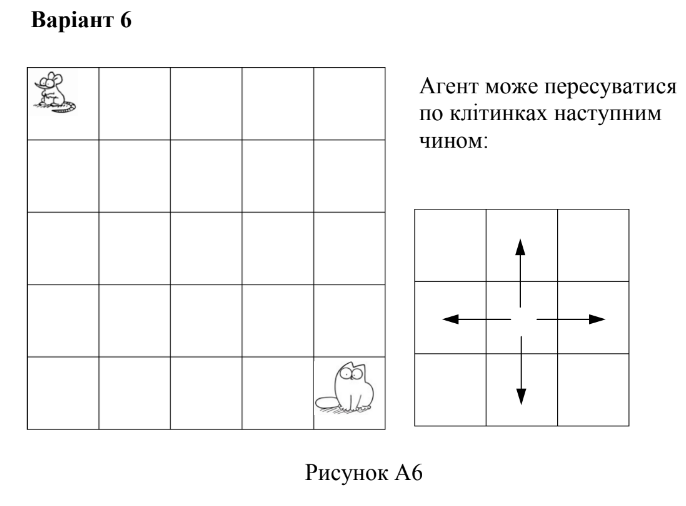
20

21

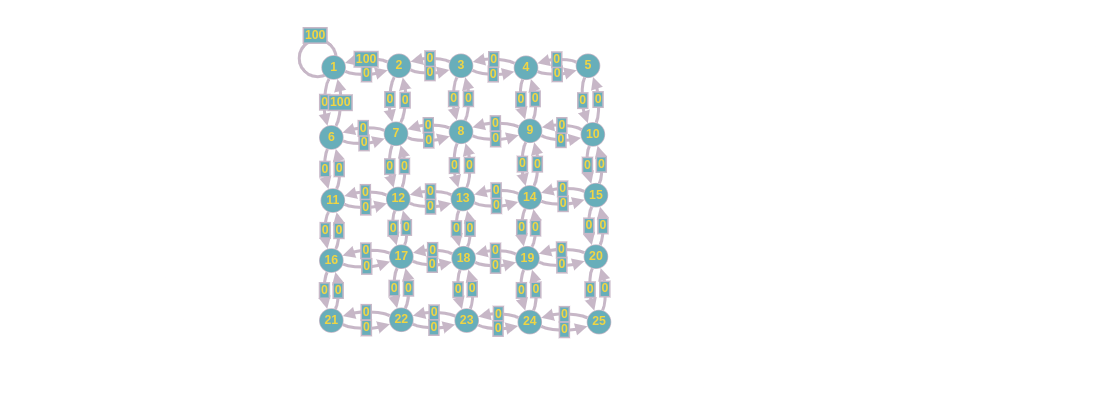
22

23

24



На основі цього малюнку побудуємо граф R



Цей граф зображує можливі ходи нашого котика за допомогою позицій(кружків) та ходів(стрілок). Винагороди зображені на стрілках. Надалі ми користуватимося таблицями R та Q.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| R | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| 1 | **100** | 0 | -1 | -1 | -1 | **0** | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| 2 | **100** | -1 | **0** | -1 | -1 | -1 | **0** | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| 3 | -1 | **0** | -1 | **0** | -1 | -1 | -1 | **0** | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| 4 | -1 | -1 | **0** | -1 | **0** | -1 | -1 | -1 | **0** | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| 5 | -1 | -1 | -1 | **0** | -1 | -1 | -1 | -1 | -1 | **0** | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| 6 | **100** | -1 | -1 | -1 | -1 | -1 | **0** | -1 | -1 | -1 | **0** | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| 7 | -1 | **0** | -1 | -1 | -1 | **0** | -1 | **0** | -1 | -1 | -1 | **0** | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| 8 | -1 | -1 | **0** | -1 | -1 | -1 | **0** | -1 | **0** | -1 | -1 | -1 | **0** | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| 9 | -1 | -1 | -1 | **0** | -1 | -1 | -1 | **0** | -1 | **0** | -1 | -1 | -1 | **0** | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| 10 | -1 | -1 | -1 | -1 | **0** | -1 | -1 | -1 | **0** | -1 | -1 | -1 | -1 | -1 | **0** | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| 11 | -1 | -1 | -1 | -1 | -1 | **0** | -1 | -1 | -1 | -1 | -1 | **0** | -1 | -1 | -1 | **0** | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| 12 | -1 | -1 | -1 | -1 | -1 | -1 | **0** | -1 | -1 | -1 | **0** | -1 | **0** | -1 | -1 | -1 | **0** | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| 13 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | **0** | -1 | -1 | -1 | **0** | -1 | **0** | -1 | -1 | -1 | **0** | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| 14 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | **0** | -1 | -1 | -1 | **0** | -1 | **0** | -1 | -1 | -1 | **0** | -1 | -1 | -1 | -1 | -1 | -1 |
| 15 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | **0** | -1 | -1 | -1 | **0** | -1 | -1 | -1 | -1 | -1 | **0** | -1 | -1 | -1 | -1 | -1 |
| 16 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | **0** | -1 | -1 | -1 | -1 | -1 | **0** | -1 | -1 | -1 | **0** | -1 | -1 | -1 | -1 |
| 17 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | **0** | -1 | -1 | -1 | **0** | -1 | **0** | -1 | -1 | -1 | **0** | -1 | -1 | -1 |
| 18 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | **0** | -1 | -1 | -1 | **0** | -1 | **0** | -1 | -1 | -1 | **0** | -1 | -1 |
| 19 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | **0** | -1 | -1 | -1 | **0** | -1 | **0** | -1 | -1 | -1 | **0** | -1 |
| 20 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | **0** | -1 | -1 | -1 | **0** | -1 | -1 | -1 | -1 | -1 | **0** |
| 21 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | **0** | -1 | -1 | -1 | -1 | -1 | **0** | -1 | -1 | -1 |
| 22 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | **0** | -1 | -1 | -1 | **0** | -1 | **0** | -1 | -1 |
| 23 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | **0** | -1 | -1 | -1 | **0** | -1 | **0** | -1 |
| 24 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | **0** | -1 | -1 | -1 | **0** | -1 | **0** |
| 25 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | **0** | -1 | -1 | -1 | **0** | -1 |

Початковий вигляд таблиці Q

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Q | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| 1 | **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 |
| 2 | **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 |
| 3 | 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 |
| 4 | 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 |
| 5 | 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 |
| 6 | **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 |
| 7 | 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 |
| 8 | 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 |
| 9 | 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 |
| 10 | 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 |
| 11 | 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 |
| 12 | 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 |
| 13 | 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 |
| 14 | 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 |
| 15 | 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 |
| 16 | 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 |
| 17 | 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 |
| 18 | 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 |
| 19 | 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 |
| 20 | 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 |
| 21 | 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 |
| 22 | 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 |
| 23 | 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 |
| 24 | 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 |
| 25 | 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 |

Тож, виходячи з того, що початок знаходиться на позиції 25, а кінець на позиції 1, проілюструємо роботу алгоритму Q-навчання розрахунками без

програмної реалізації в 2ох діях.

Рухи кота

14

2

4

3

6^

7

12

11^

10

5

9

8

13

1.

15

16^

17

18

19

20

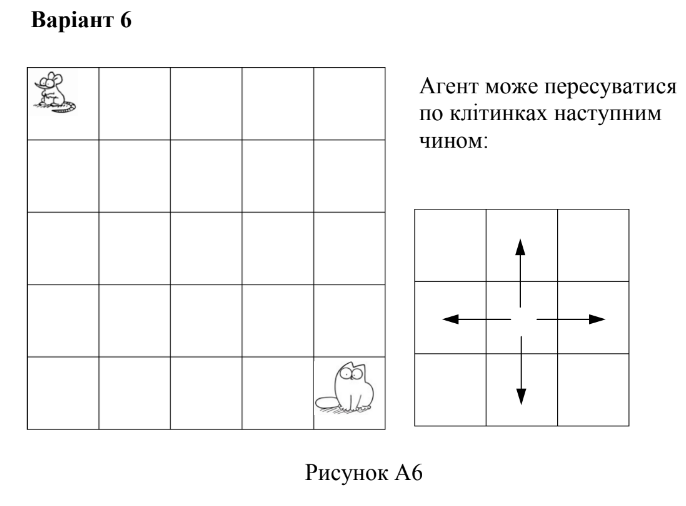
21^

<-22

<-23

<-24

<-25



|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Перша дія** | | | | | | | | | |  |
| G = 0.8 |  |  |  |  |  | |  | |  | |
| Крок = 1 | Випадково йдемо з 25 в 24 Q[25,24] | | |  |  | |  | |  | |
| - | - | |  |  |  | |  | |  | |
| Крок = 2 | Випадково йдемо з 24 в 23 Q[24,23] | | |  |  | |  | |  | |
| Q[25,24] = | R[25,24] + G \* MAX(Q[24,19];Q[24,23]) = 0 + 0.8 \* 0 = 0 | | | | | | | | |  |
|  |  |  |  |  |  | |  | |  | |
| Крок = 3 | Випадково йдемо з 23 в 22 Q[23,22] | | |  |  | |  | |  | |
| Q[24,23] = | R[24,23] + G \* MAX(Q[23,22];Q[23,18];Q[23,24]) = 0 + 0.8 \* 0 = 0 | | | | | | | | |  |
|  |  |  |  |  |  | |  | |  | |
| Крок = 4 | Випадково йдемо з 22 в 21 Q[22,21] | | |  |  | |  | |  | |
| Q[23,22] = | R[23,22] + G \* MAX(Q[22,21];Q[22,17];Q[22,23]) = 0 + 0.8 \* 0 = 0 | | | | | | | | |  |
|  |  |  |  |  |  | |  | |  | |
| Крок = 5 | Випадково йдемо з 21 в 16 Q[21,16] | | |  |  | |  | |  | |
| Q[22,21] = | R[22,21] + G \* MAX(Q[21,16];Q[21,22]) = 0 + 0.8 \* 0 = 0 | | | | | | | | |  |
|  |  |  |  |  |  | |  | |  | |
| Крок = 6 | Випадково йдемо з 16 в 11 Q[16,11] | | |  |  | |  | |  | |
| Q[21,16] = | R[21,16] + G \* MAX(Q[16,11];Q[16,17];Q[16,21]) = 0 + 0.8 \* 0 = 0 | | | | | | | | |  |
|  |  |  |  |  |  | |  | |  | |
| Крок = 7 | Випадково йдемо з 11 в 6 Q[11,6] | | |  |  | |  | |  | |
| Q[16,11] = | R[16,11] + G \* MAX(Q[11,6];Q[11,12];Q[11,26]) = 0 + 0.8 \* 0 = 0 | | | | | | | | |  |
|  |  |  |  |  |  | |  | |  | |
| **Крок = 8** | **Змушено йдемо з 6 в 1 Q[6,1]** | | | | |  | |  | |  |
| **Q[11,6] =** | **R[11,6] + G \* MAX(Q[6,1];Q[6,7];Q[6,11]) = 0 + 0.8 \* 100 = 80** | | | | | | | | |  |
|  |  |  |  |  |  | |  | |  | |
| **Крок = 9** | **Змушено йдемо з 1 в 1 Q[1,1]** | | | | |  | |  | |  |
| **Q[6,1] =** | **R[6,1] + G \* MAX(Q[1,6];Q[1,2]) = 100 + 0.8 \* 100 = 180** | | | | | | | | |  |
|  |  |  |  |  |  | |  | |  | |
| **Крок = 10** | **Кінець** | | | | |  | |  | |  |
| **Q[1,1] =** | **R[1,1] + G \* MAX(Q[1,6];Q[1,2];Q[1,1]) = 100 + 0.8 \* 100 = 180** | | | | | | | | |  |

Запишемо результати наших обчислень в таблицю Q

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Q | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| 1 | **180** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | **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 |
| 3 | 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 |
| 4 | 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 |
| 5 | 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 |
| 6 | **180** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 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 |
| 8 | 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 |
| 9 | 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 |
| 10 | 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 |
| 11 | 0 | 0 | 0 | 0 | 0 | **80** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 12 | 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 |
| 13 | 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 |
| 14 | 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 |
| 15 | 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 |
| 16 | 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 |
| 17 | 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 |
| 18 | 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 |
| 19 | 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 |
| 20 | 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 |
| 21 | 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 |
| 22 | 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 |
| 23 | 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 |
| 24 | 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 |
| 25 | 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 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Друга дія** | | | | | | | |
| G = 0.8 |  |  |  |  |  |  |  |
| Крок = 1 | випадково йдемо з 25 йдемо в 24 Q[25,24] | | | | | |  |
| - | - | |  |  |  |  |  |
| Крок = 2 | випадково йдемо з 24 йдемо в 23 Q[24,23] | | | | | |  |
| Q[25,24] = | R[25,24] + G \* MAX(Q[24,19];Q[24,23]) = 0 + 0.8 \* 0 = 0 | | | | | | |
|  |  |  |  |  |  |  |  |
| Крок = 3 | випадково йдемо з 23 йдемо в 22 Q[23,22] | | | | | |  |
| Q[24,23] = | R[24,23] + G \* MAX(Q[23,22];Q[23,18];Q[23,24]) = 0 + 0.8 \* 0 = 0 | | | | | | |
|  |  |  |  |  |  |  |  |
| Крок = 4 | випадково йдемо з 22 йдемо в 21 Q[22,21] | | | | | |  |
| Q[23,22] = | R[23,22] + G \* MAX(Q[22,21];Q[22,17];Q[22,23]) = 0 + 0.8 \* 0 = 0 | | | | | | |
|  |  |  |  |  |  |  |  |
| Крок = 5 | випадково йдемо з 21 йдемо в 16 Q[21,16] | | | | | |  |
| Q[22,21] = | R[22,21] + G \* MAX(Q[21,16];Q[21,22]) = 0 + 0.8 \* 0 = 0 | | | | | | |
|  |  |  |  |  |  |  |  |
| Крок = 6 | випадково йдемо з 16 йдемо в 11 Q[16,11] | | | | | |  |
| Q[21,16] = | R[21,16] + G \* MAX(Q[16,11];Q[16,17];Q[16,21]) = 0 + 0.8 \* 0 = 0 | | | | | | |
|  |  |  |  |  |  |  |  |
| **Крок = 7** | **змушено йдемо з 11 в 6 Q[11,6]** | | | |  |  |  |
| **Q[16,11] =** | **R[16,11] + G \* MAX(Q[11,6];Q[11,12];Q[11,26]) = 0 + 0.8 \* 80 = 64** | | | | | | |
|  |  |  |  |  |  |  |  |
| **Крок = 8** | **змушено йдемо з 6 в 1 Q[6,1]** | | | | |  |  |
| **Q[11,6] =** | **R[11,6] + G \* MAX(Q[6,1];Q[6,7];Q[6,11]) = 0 + 0.8 \* 180 = 144** | | | | | | |
|  |  |  |  |  |  |  |  |
| **Крок = 9** | **змушено йдемо з 1 в 1 Q[1,1]** | | | | |  |  |
| **Q[6,1] =** | **R[6,1] + G \* MAX(Q[1,6];Q[1,2]) = 100 + 0.8 \* 180 = 244** | | | | | | |
|  |  |  |  |  |  |  |  |
| **Крок = 10** | **Кінець** | | | | |  |  |
| **Q[1,1] =** | **R[1,1] + G \* MAX(Q[1,6];Q[1,2];Q[1,1) = 100 + 0.8 \* 180 = 244** | | | | | | |

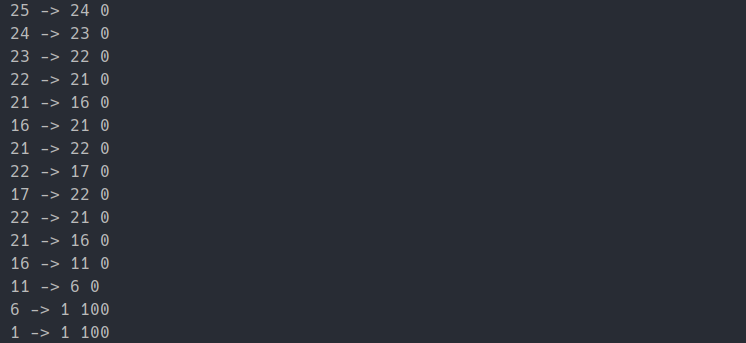
Тож таблиця Q матиме вигляд:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Q | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| 1 | **244** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | **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 |
| 3 | 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 |
| 4 | 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 |
| 5 | 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 |
| 6 | **244** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 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 |
| 8 | 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 |
| 9 | 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 |
| 10 | 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 |
| 11 | 0 | 0 | 0 | 0 | 0 | **144** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 12 | 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 |
| 13 | 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 |
| 14 | 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 |
| 15 | 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 |
| 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **64** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 17 | 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 |
| 18 | 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 |
| 19 | 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 |
| 20 | 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 |
| 21 | 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 |
| 22 | 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 |
| 23 | 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 |
| 24 | 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 |
| 25 | 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 |

**Завдання 2**

Програмна реалізація складається з отримання даних з ексель файлу source. Отримаємо початкові таблиці Q та R за допомогою бібліотеки pandas. Та результат - таблиця Q - виводимо в ексель.

Далі запускаємо кота пройтися n раз по коридорам.

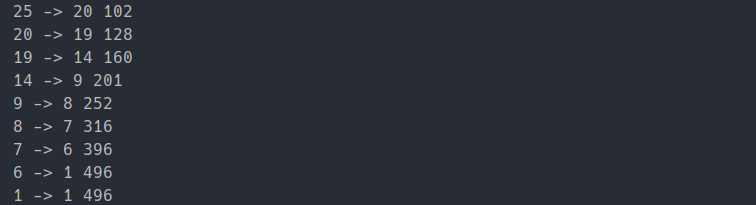


Можна помітити, що кіт безцільно блукає оскільки не пам’ятає шлях до цілі в першій епосі. Більше того, він може вертатися на ту позицію де вже був, що не є ефективною стратегією.

Далі з кожною ітерацією кіт все краще й краще запам’ятовує маршрут. Для запам`ятовування доріг в таблицю Q використовується формула:

Q[pos][next\_pos] = R[pos][next\_pos] + G \* Q[next\_pos][get\_next\_pos\_index(Q, R, next\_pos)]

Тож вже на 50 поході на мишу кіт знаходить один з можливих найкоротших шляхів, що складає 9 рухів :



Таблиця Q матиме вигляд :

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| 1 | **496** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | **244** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 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 |
| 4 | 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 |
| 5 | 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 |
| 6 | **496** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | **396** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | **316** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **252** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 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 |
| 11 | 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 |
| 12 | 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 |
| 13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **51** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **201** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 15 | 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 |
| 16 | 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 |
| 17 | 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 |
| 18 | 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 |
| 19 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **160** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **128** | 0 | 0 | 0 | 0 | 0 | 0 |
| 21 | 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 |
| 22 | 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 |
| 23 | 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 |
| 24 | 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 |
| 25 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **102** | 0 | 0 | 0 | 0 | 0 |

**Код**

main.py

*import* pandas *as* pd  
*from* movement *import* \*  
pos = 25

R = pd.read\_excel('source.xlsx', sheet\_name='R').set\_index('R')  
Q = pd.read\_excel('source.xlsx', sheet\_name='Q\_start').set\_index('Q')

*for* i *in* range(50):  
 Q = move(np.array(Q), np.array(R), -1, pos - 1)  
 print(Q)

df = pd.DataFrame(np.array(Q))  
df.index += 1  
df.columns += 1  
df.to\_excel("Q\_table.xlsx")

movement.py

*import* numpy *as* np  
  
G = 0.8

*def* move(Q, R, prev\_pos, pos):  
 *if* prev\_pos == pos: *return* Q  
 next\_pos = get\_next\_pos\_index(Q, R, pos)  
 Q[pos][next\_pos] = R[pos][next\_pos] + G \* Q[next\_pos][get\_next\_pos\_index(Q, R, next\_pos)]  
 print(pos + 1, "->", next\_pos + 1, Q[pos][next\_pos])  
 *return* move(Q, R, pos, next\_pos)  
  
  
*def* get\_max\_value(matrix, pas, pos):  
 paths = np.where(matrix[pos] > pas)[0]  
 *if* len(paths) == 0: *return* -1  
 path\_value = [matrix[pos][i] *for* i *in* paths]  
 *return* np.amax(path\_value)  
  
  
*def* get\_indexes\_by\_value(Q, R, pos):  
 max\_value\_q = get\_max\_value(Q, 0, pos)  
 *if* max\_value\_q != -1: *return* np.where(Q[pos] == max\_value\_q)  
 max\_value\_r = get\_max\_value(R, -1, pos)  
 *return* np.where(R[pos] == max\_value\_r)  
  
  
*def* get\_next\_pos\_index(Q, R, pos):  
 max\_indexes = get\_indexes\_by\_value(Q, R, pos)  
 *return* np.random.choice(max\_indexes[0])

Висновки

В ході лабораторної роботи було розроблено розумного робота, що змозі досягти цілі за найменшу кількість кроків. Був вивчений та застосований алгоритм Q навчання. Спершу бот блукає, оскільки не знає маршрут до цілі. В основному він спочатку орієнтується на таблицю R - таблиця доступних рухів . З кожною епохою бот все детальніше запам’ятовує дорогу. Його пам’ять це таблиця Q й з кожною епохою бот все частіше користується саме нею для наближення до цілі. Ця таблиця заповнюється згідно формули оновлення таблиці Q:

Q[s, a] = R[s, a] + Gamma \* MAX(Q[s', a']),

де Q[s, a] – комірка матриці Q, що відповідає поточному стану агента;

R[s, a] – комірка матриці R, що відповідає поточному стану агента;

Gamma – швидкість навчання, рекомендоване значення = 0.8;

Q[s', a'] – комірка матриці Q, що відповідає наступному стану агента;

MAX(Q[s', a']) – вибір з множини можливих дій агента в поточному

стані дії з максимальною винагородою.