



Department of Electronics Engineering
National Chiao Tung University
Hsinchu, Taiwan

Programming assignment #2



Student:

Chieh Min Yu

Advisor:

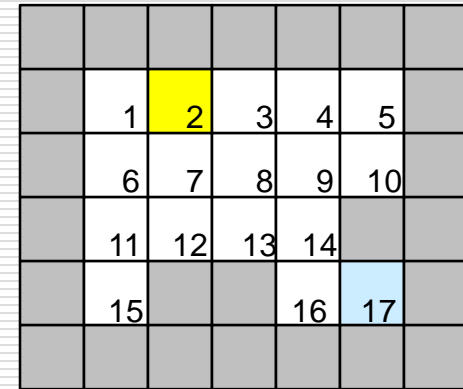
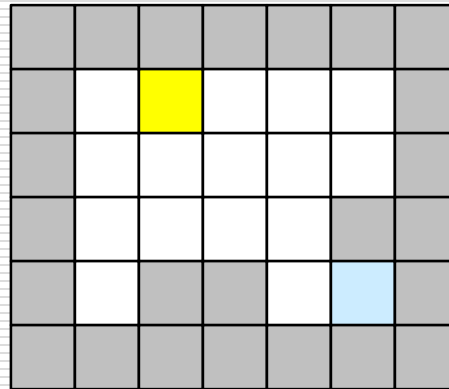
Juinn-Dar Huang, Ph. D.

October 02, 2018

Input file example ↵

```

7 ↵
6 ↵
2 2 2 2 2 2 2 ↵
2 0 S 0 0 0 2 ↵
2 0 0 0 0 0 2 ↵
2 0 0 0 0 2 2 ↵
2 0 2 2 0 E 2 ↵
2 2 2 2 2 2 2 ↵
  
```



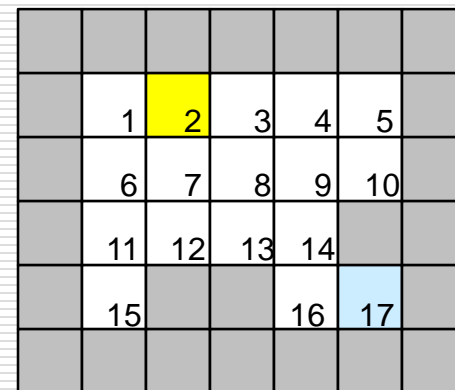
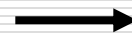
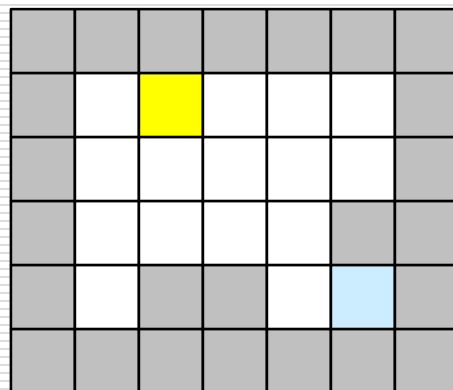
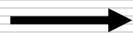
給編號

Input file example ↴

```

7 ↴
6 ↴
2 2 2 2 2 2 2 ↴
2 0 S 0 0 0 2 ↴
2 0 0 0 0 0 2 ↴
2 0 0 0 0 2 2 ↴
2 0 2 2 0 E 2 ↴
2 2 2 2 2 2 2 ↴

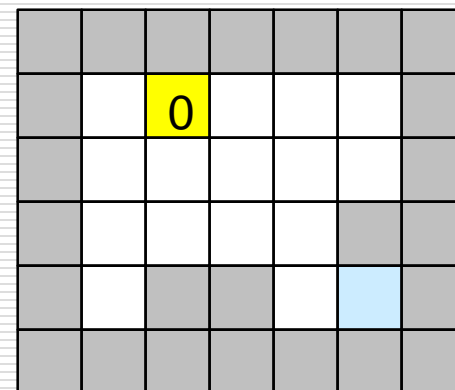
```



給編號

方法二其中一種解法： ↴

- ➔ (1) 將起點存進queue，並記錄距離(與起點的距離)為0 ↴
- (2) 取出queue中的第一個“儲存的位置(P)”，且其紀錄的距離為d ↴
- (3) 將P周圍“尚未存進過queue裡面的”位置存到queue裡面，並記錄距離為d+1 ↴
- (4) 重複(2)、(3)直到找到終點 ↴
- 從起點到終點最少需要的步數，為終點所記錄的距離(例如:終點與起點有k步的距離)；而起點到終點的路徑則是，從終點往回找 -> 距離為(k-1)的位置 -> 距離為(k-2)的位置 -> -> 距離為1的位置 -> 起點。 ↴

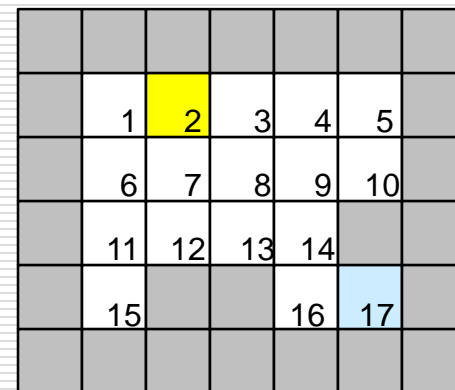
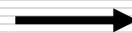
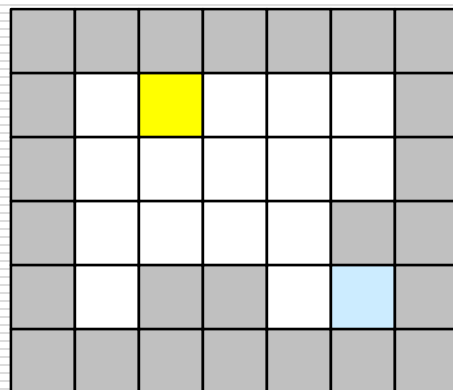


紀錄距離

queue 2

Input file example ↴

```
7 ↴
6 ↴
2 2 2 2 2 2 2 ↴
2 0 S 0 0 0 2 ↴
2 0 0 0 0 0 2 ↴
2 0 0 0 0 2 2 ↴
2 0 2 2 0 E 2 ↴
2 2 2 2 2 2 2 ↴
↴
```



給編號

方法二其中一種解法：↴

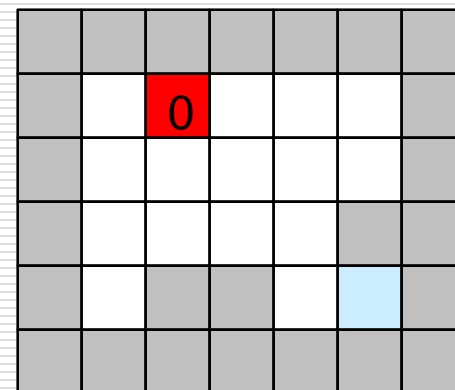
(1) 將起點存進queue，並記錄距離(與起點的距離)為0 ↴

→ (2) 取出queue中的第一個“儲存的位置(P)”，且其紀錄的距離為d ↴

(3) 將P周圍“尚未存進過queue裡面的”位置存到queue裡面，並記錄距離為d+1 ↴

(4) 重複(2)、(3)直到找到終點 ↴

從起點到終點最少需要的步數，為終點所記錄的距離(例如:終點與起點有k步的距離)；而起點到終點的路徑則是，從終點往回找 -> 距離為(k-1)的位置 -> 距離為(k-2)的位置 -> -> 距離為1的位置 -> 起點。↴



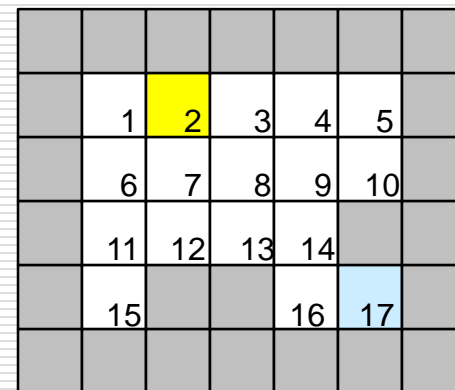
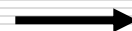
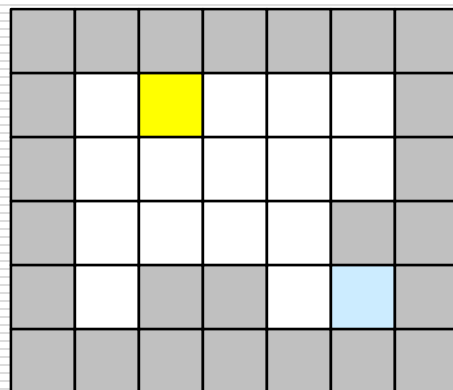
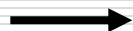
紀錄距離

$P = 2, d = 0$

queue 2

Input file example ↴

```
7 ↴
6 ↴
2 2 2 2 2 2 2 ↴
2 0 S 0 0 0 2 ↴
2 0 0 0 0 0 2 ↴
2 0 0 0 0 2 2 ↴
2 0 2 2 0 E 2 ↴
2 2 2 2 2 2 2 ↴
↴
```



給編號

方法二其中一種解法：↴

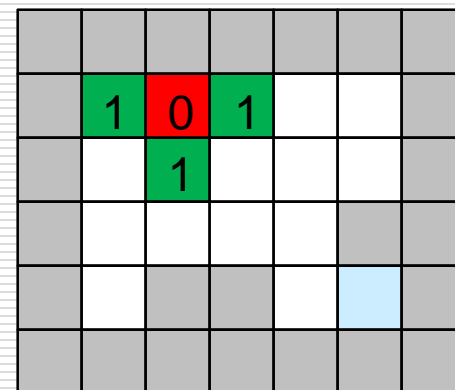
(1) 將起點存進queue，並記錄距離(與起點的距離)為0 ↴

(2) 取出queue中的第一個"儲存的位置(P)"，且其紀錄的距離為d ↴

→ (3) 將P周圍"尚未存進過queue裡面的"位置存到queue裡面，並記錄距離為d+1 ↴

(4) 重複(2)、(3)直到找到終點 ↴

從起點到終點最少需要的步數，為終點所記錄的距離(例如:終點與起點有k步的距離)；而起點到終點的路徑則是，從終點往回找 -> 距離為(k-1)的位置 -> 距離為(k-2)的位置 -> -> 距離為1的位置 -> 起點。↴



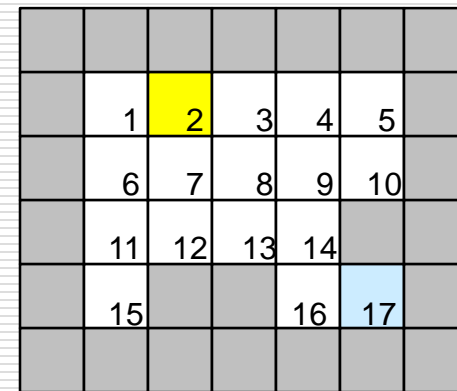
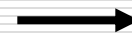
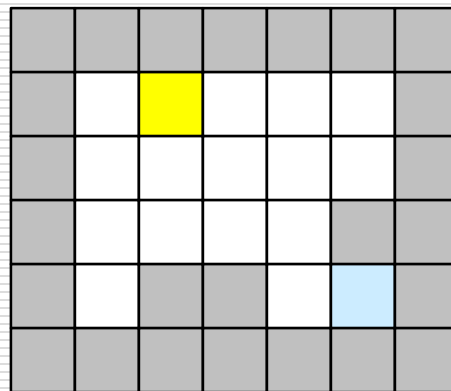
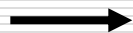
紀錄距離

$P = 2, d = 0$

queue 1、3、7

Input file example ↴

```
7 ↴
6 ↴
2 2 2 2 2 2 2 ↴
2 0 S 0 0 0 2 ↴
2 0 0 0 0 0 2 ↴
2 0 0 0 0 2 2 ↴
2 0 2 2 0 E 2 ↴
2 2 2 2 2 2 2 ↴
↴
```



給編號

方法二其中一種解法：↴

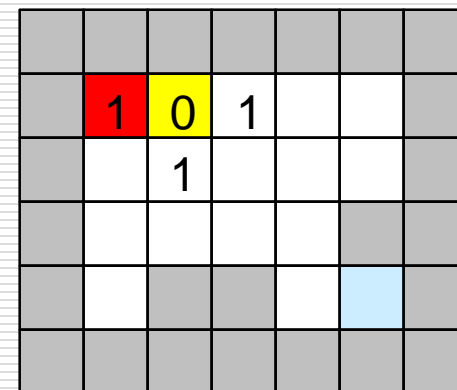
(1) 將起點存進queue，並記錄距離(與起點的距離)為0。↴

→ (2) 取出queue中的第一個“儲存的位置(P)”，且其紀錄的距離為d。↴

(3) 將P周圍“尚未存進過queue裡面的”位置存到queue裡面，並記錄距離為d+1。↴

(4) 重複(2)、(3)直到找到終點。↴

從起點到終點最少需要的步數，為終點所記錄的距離(例如:終點與起點有k步的距離)；而起點到終點的路徑則是，從終點往回找 -> 距離為(k-1)的位置 -> 距離為(k-2)的位置 -> -> 距離為1的位置 -> 起點。↴



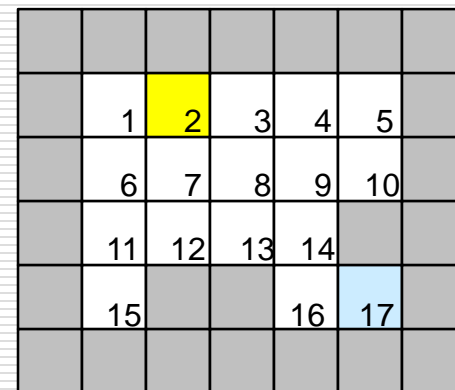
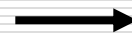
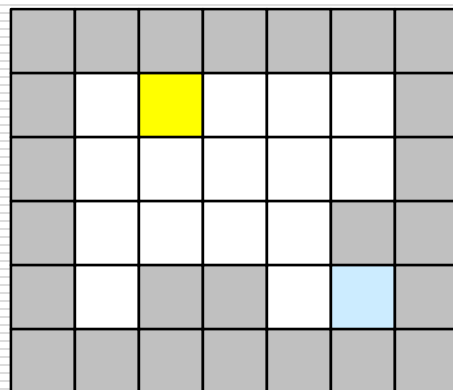
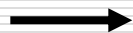
紀錄距離

P = 1、d = 1

queue 1、3、7

Input file example ↴

```
7 ↴
6 ↴
2 2 2 2 2 2 2 ↴
2 0 S 0 0 0 2 ↴
2 0 0 0 0 0 2 ↴
2 0 0 0 0 2 2 ↴
2 0 2 2 0 E 2 ↴
2 2 2 2 2 2 2 ↴
↴
```



給編號

方法二其中一種解法：↴

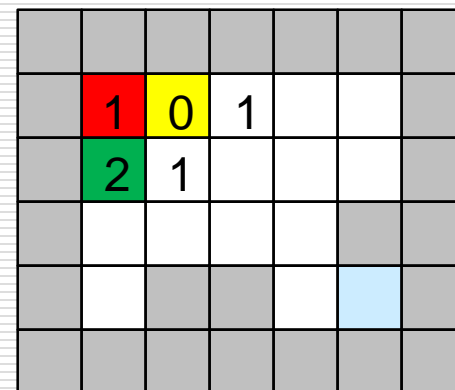
(1) 將起點存進queue，並記錄距離(與起點的距離)為0 ↴

(2) 取出queue中的第一個"儲存的位置(P)"，且其紀錄的距離為d ↴

→ (3) 將P周圍"尚未存進過queue裡面的"位置存到queue裡面，並記錄距離為d+1 ↴

(4) 重複(2)、(3)直到找到終點 ↴

從起點到終點最少需要的步數，為終點所記錄的距離(例如:終點與起點有k步的距離)；而起點到終點的路徑則是，從終點往回找 -> 距離為(k-1)的位置 -> 距離為(k-2)的位置 -> -> 距離為1的位置 -> 起點。↴



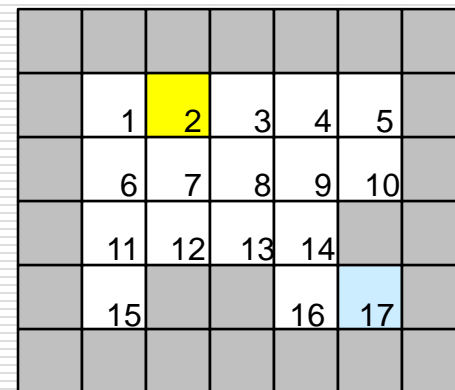
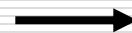
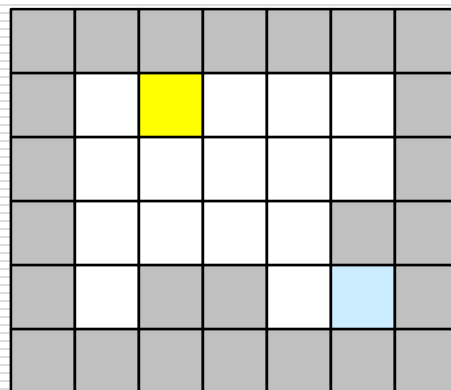
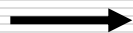
紀錄距離

P = 1、d = 1

queue 3、7、6

Input file example ↴

```
7 ↴
6 ↴
2 2 2 2 2 2 2 ↴
2 0 S 0 0 0 2 ↴
2 0 0 0 0 0 2 ↴
2 0 0 0 0 2 2 ↴
2 0 2 2 0 E 2 ↴
2 2 2 2 2 2 2 ↴
↴
```



給編號

方法二其中一種解法：↴

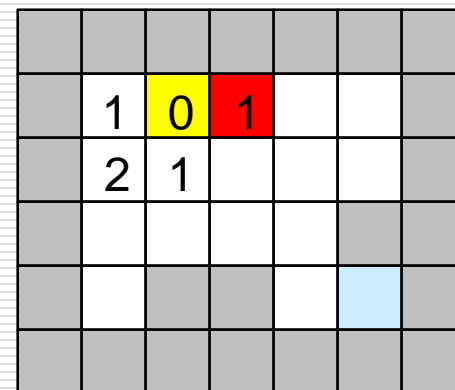
(1) 將起點存進queue，並記錄距離(與起點的距離)為0 ↴

→ (2) 取出queue中的第一個“儲存的位置(P)”，且其紀錄的距離為d ↴

(3) 將P周圍“尚未存進過queue裡面的”位置存到queue裡面，並記錄距離為d+1 ↴

(4) 重複(2)、(3)直到找到終點 ↴

從起點到終點最少需要的步數，為終點所記錄的距離(例如:終點與起點有k步的距離)；而起點到終點的路徑則是，從終點往回找 -> 距離為(k-1)的位置 -> 距離為(k-2)的位置 -> -> 距離為1的位置 -> 起點。↴



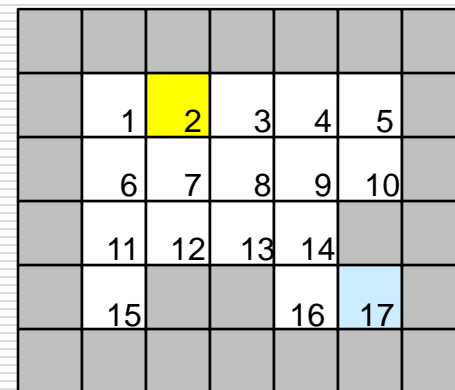
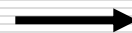
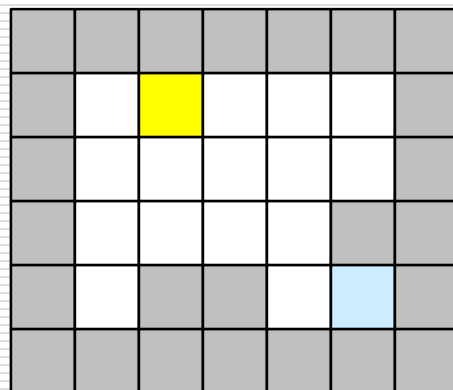
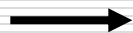
紀錄距離

P = 3、d = 1

queue 3、7、6

Input file example ↴

```
7 ↴
6 ↴
2 2 2 2 2 2 2 ↴
2 0 S 0 0 0 2 ↴
2 0 0 0 0 0 2 ↴
2 0 0 0 0 2 2 ↴
2 0 2 2 0 E 2 ↴
2 2 2 2 2 2 2 ↴
↴
```



給編號

方法二其中一種解法：↴

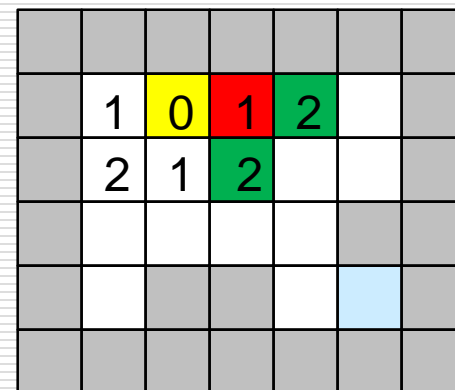
(1) 將起點存進queue，並記錄距離(與起點的距離)為0↴

(2) 取出queue中的第一個"儲存的位置(P)"，且其紀錄的距離為d↴

→ (3) 將P周圍"尚未存進過queue裡面的"位置存到queue裡面，並記錄距離為d+1↴

(4) 重複(2)、(3)直到找到終點↴

從起點到終點最少需要的步數，為終點所記錄的距離(例如:終點與起點有k步的距離)；而起點到終點的路徑則是，從終點往回找 -> 距離為(k-1)的位置 -> 距離為(k-2)的位置 -> -> 距離為1的位置 -> 起點。↴



紀錄距離

P = 3、d = 1

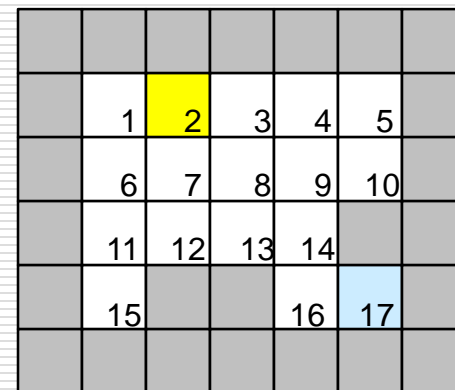
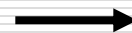
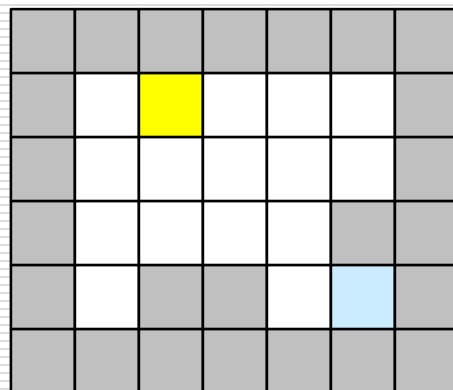
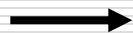
queue 7、6、8、4

Input file example ↴

```

7 ↴
6 ↴
2 2 2 2 2 2 2 ↴
2 0 S 0 0 0 2 ↴
2 0 0 0 0 0 2 ↴
2 0 0 0 0 2 2 ↴
2 0 2 2 0 E 2 ↴
2 2 2 2 2 2 2 ↴

```



給編號

方法二其中一種解法： ↴

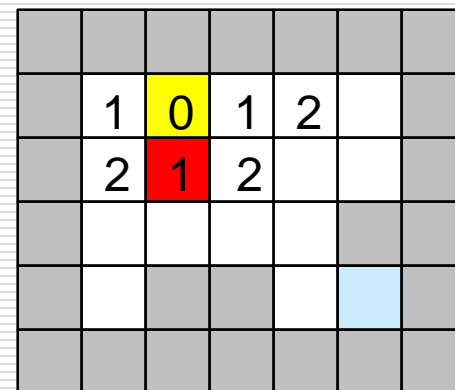
(1) 將起點存進queue，並記錄距離(與起點的距離)為0 ↴

→ (2) 取出queue中的第一個“儲存的位置(P)”，且其紀錄的距離為d ↴

(3) 將P周圍“尚未存進過queue裡面的”位置存到queue裡面，並記錄距離為d+1 ↴

(4) 重複(2)、(3)直到找到終點 ↴

從起點到終點最少需要的步數，為終點所記錄的距離(例如:終點與起點有k步的距離)；而起點到終點的路徑則是，從終點往回找 -> 距離為(k-1)的位置 -> 距離為(k-2)的位置 -> -> 距離為1的位置 -> 起點。 ↴



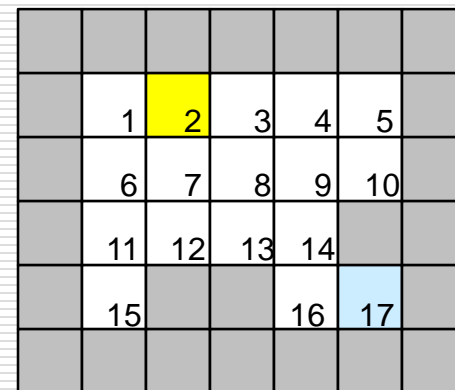
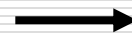
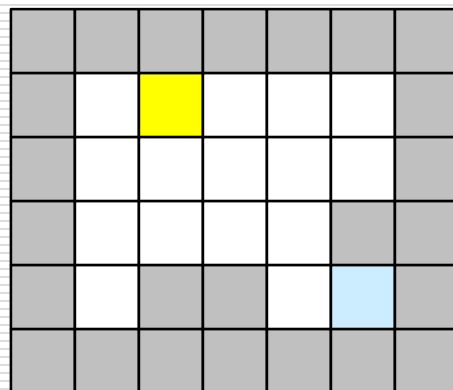
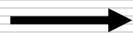
紀錄距離

P = 7、d = 1

queue 7、6、8、4

Input file example ↴

```
7 ↴
6 ↴
2 2 2 2 2 2 2 ↴
2 0 S 0 0 0 2 ↴
2 0 0 0 0 0 2 ↴
2 0 0 0 0 2 2 ↴
2 0 2 2 0 E 2 ↴
2 2 2 2 2 2 2 ↴
↴
```



給編號

方法二其中一種解法：↴

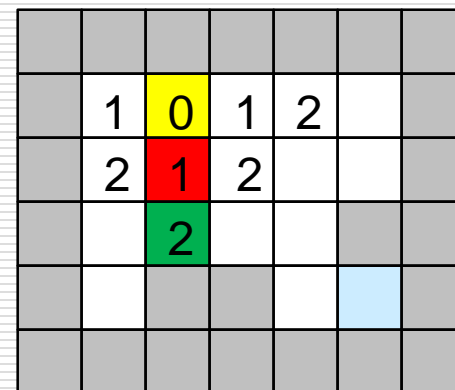
(1) 將起點存進queue，並記錄距離(與起點的距離)為0 ↴

(2) 取出queue中的第一個“儲存的位置(P)”，且其紀錄的距離為d ↴

→ (3) 將P周圍“尚未存進過queue裡面的”位置存到queue裡面，並記錄距離為d+1 ↴

(4) 重複(2)、(3)直到找到終點 ↴

從起點到終點最少需要的步數，為終點所記錄的距離(例如:終點與起點有k步的距離)；而起點到終點的路徑則是，從終點往回找 -> 距離為(k-1)的位置 -> 距離為(k-2)的位置 -> -> 距離為1的位置 -> 起點。↴



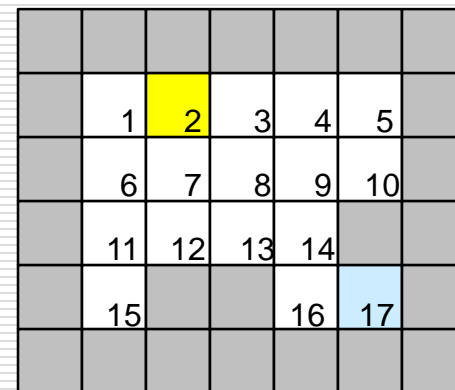
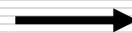
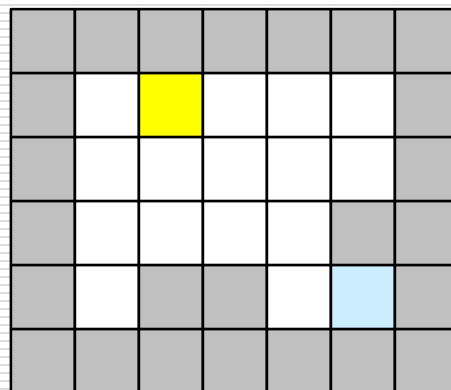
紀錄距離

P = 7、d = 1

queue 6、8、4、12

Input file example ↴

```
7 ↴
6 ↴
2 2 2 2 2 2 2 ↴
2 0 S 0 0 0 2 ↴
2 0 0 0 0 0 2 ↴
2 0 0 0 0 2 2 ↴
2 0 2 2 0 E 2 ↴
2 2 2 2 2 2 2 ↴
↴
```



給編號

方法二其中一種解法：↴

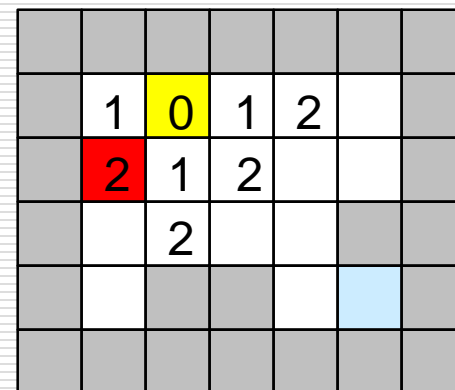
(1) 將起點存進queue，並記錄距離(與起點的距離)為0 ↴

→ (2) 取出queue中的第一個“儲存的位置(P)”，且其紀錄的距離為d ↴

(3) 將P周圍“尚未存進過queue裡面的”位置存到queue裡面，並記錄距離為d+1 ↴

(4) 重複(2)、(3)直到找到終點 ↴

從起點到終點最少需要的步數，為終點所記錄的距離(例如:終點與起點有k步的距離)；而起點到終點的路徑則是，從終點往回找 -> 距離為(k-1)的位置 -> 距離為(k-2)的位置 -> -> 距離為1的位置 -> 起點。↴



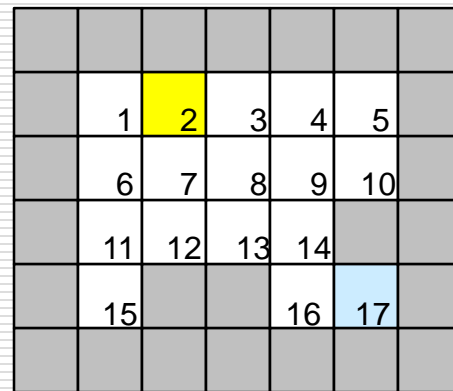
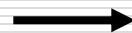
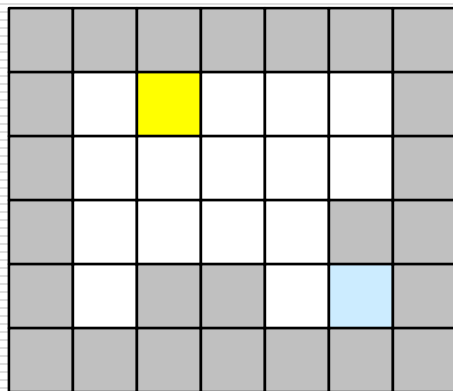
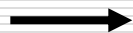
紀錄距離

P = 6、d = 2

queue 6、8、4、12

Input file example ↴

```
7 ↴
6 ↴
2 2 2 2 2 2 2 ↴
2 0 S 0 0 0 2 ↴
2 0 0 0 0 0 2 ↴
2 0 0 0 0 2 2 ↴
2 0 2 2 0 E 2 ↴
2 2 2 2 2 2 2 ↴
↴
```



給編號

方法二其中一種解法：↴

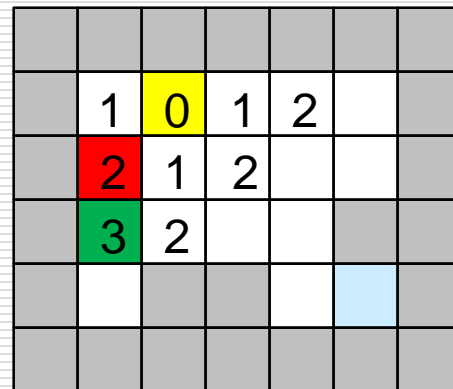
(1) 將起點存進queue，並記錄距離(與起點的距離)為0 ↴

(2) 取出queue中的第一個“儲存的位置(P)”，且其紀錄的距離為d ↴

➡ (3) 將P周圍“尚未存進過queue裡面的”位置存到queue裡面，並記錄距離為d+1 ↴

(4) 重複(2)、(3)直到找到終點 ↴

從起點到終點最少需要的步數，為終點所記錄的距離(例如:終點與起點有k步的距離)；而起點到終點的路徑則是，從終點往回找 -> 距離為(k-1)的位置 -> 距離為(k-2)的位置 -> -> 距離為1的位置 -> 起點。↴



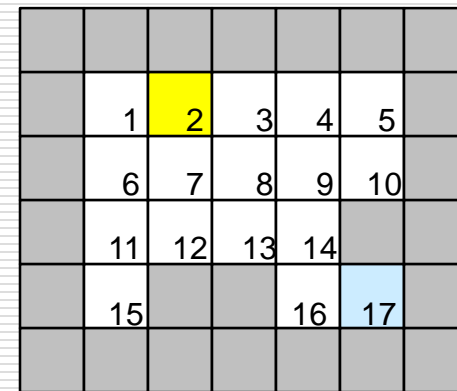
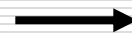
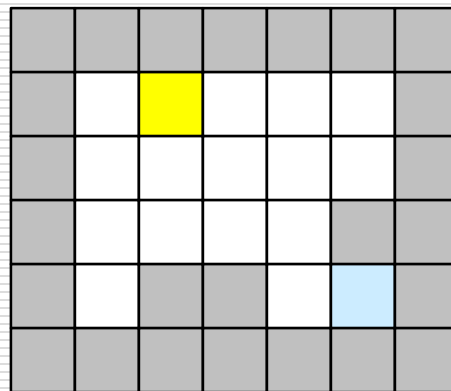
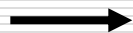
紀錄距離

P = 6、d = 2

queue 8、4、12、11

Input file example ↴

```
7 ↴
6 ↴
2 2 2 2 2 2 2 ↴
2 0 S 0 0 0 2 ↴
2 0 0 0 0 0 2 ↴
2 0 0 0 0 2 2 ↴
2 0 2 2 0 E 2 ↴
2 2 2 2 2 2 2 ↴
↴
```

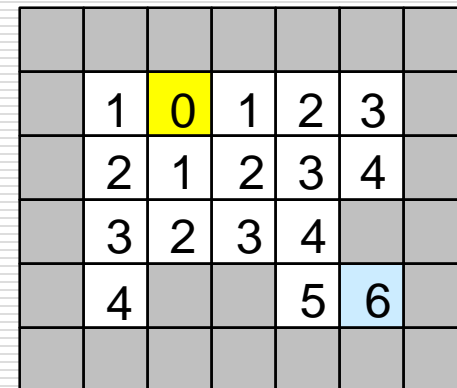


給編號

方法二其中一種解法：↴

- (1) 將起點存進queue，並記錄距離(與起點的距離)為0↴
- (2) 取出queue中的第一個"儲存的位置(P)"，且其紀錄的距離為d↴
- (3) 將P周圍"尚未存進過queue裡面的"位置存到queue裡面，並記錄距離為d+1↴
- (4) 重複(2)、(3)直到找到終點↴

從起點到終點最少需要的步數，為終點所記錄的距離(例如:終點與起點有k步的距離)；而起點到終點的路徑則是，從終點往回找 -> 距離為(k-1)的位置 -> 距離為(k-2)的位置 -> -> 距離為1的位置 -> 起點。↴



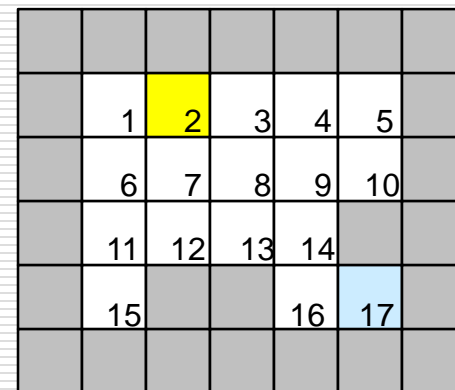
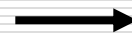
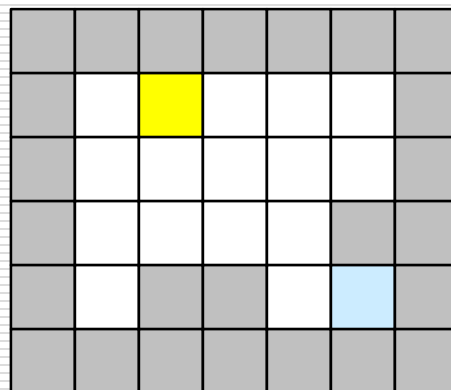
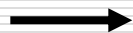
紀錄距離

$$P = 16, d = 5$$

queue 17

Input file example ↴

```
7 ↴
6 ↴
2 2 2 2 2 2 2 ↴
2 0 S 0 0 0 2 ↴
2 0 0 0 0 0 2 ↴
2 0 0 0 0 2 2 ↴
2 0 2 2 0 E 2 ↴
2 2 2 2 2 2 2 ↴
↴
```

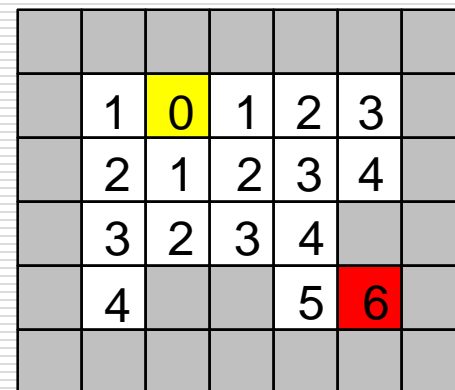


給編號

方法二其中一種解法：↴

- (1) 將起點存進queue，並記錄距離(與起點的距離)為0↴
- (2) 取出queue中的第一個"儲存的位置(P)"，且其紀錄的距離為d↴
- (3) 將P周圍"尚未存進過queue裡面的"位置存到queue裡面，並記錄距離為d+1↴
- (4) 重複(2)、(3)直到找到終點↴

從起點到終點最少需要的步數，為終點所記錄的距離(例如:終點與起點有k步的距離)；而起點到終點的路徑則是，從終點往回找 -> 距離為(k-1)的位置 -> 距離為(k-2)的位置 -> -> 距離為1的位置 -> 起點。↴



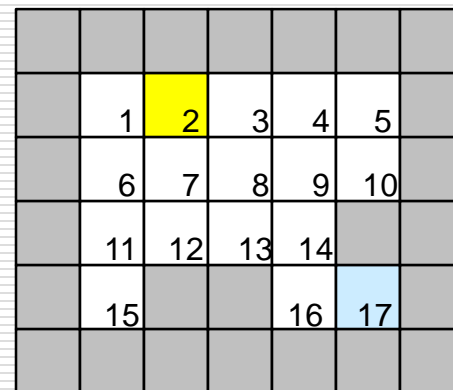
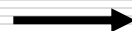
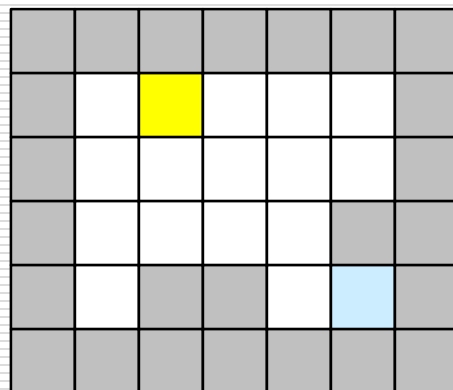
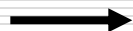
紀錄距離

$$P = 16, d = 5$$

queue 17

Input file example ↴

```
7 ↴
6 ↴
2 2 2 2 2 2 2 ↴
2 0 S 0 0 0 2 ↴
2 0 0 0 0 0 2 ↴
2 0 0 0 0 2 2 ↴
2 0 2 2 0 E 2 ↴
2 2 2 2 2 2 2 ↴
↴
```

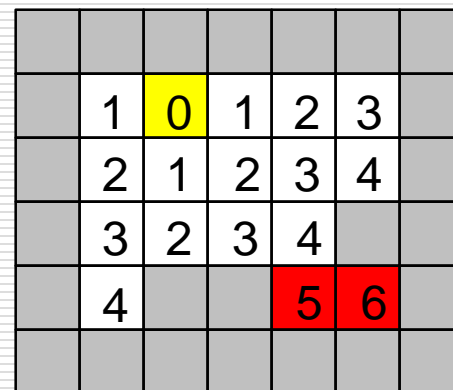


給編號

方法二其中一種解法：↴

- (1) 將起點存進queue，並記錄距離(與起點的距離)為0↴
- (2) 取出queue中的第一個"儲存的位置(P)"，且其紀錄的距離為d↴
- (3) 將P周圍"尚未存進過queue裡面的"位置存到queue裡面，並記錄距離為d+1↴
- (4) 重複(2)、(3)直到找到終點↴

從起點到終點最少需要的步數，為終點所記錄的距離(例如:終點與起點有k步的距離)；而起點到終點的路徑則是，從終點往回找 -> 距離為(k-1)的位置 -> 距離為(k-2)的位置 -> -> 距離為1的位置 -> 起點。↴



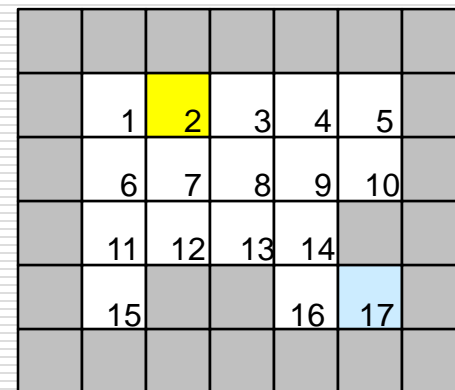
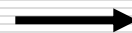
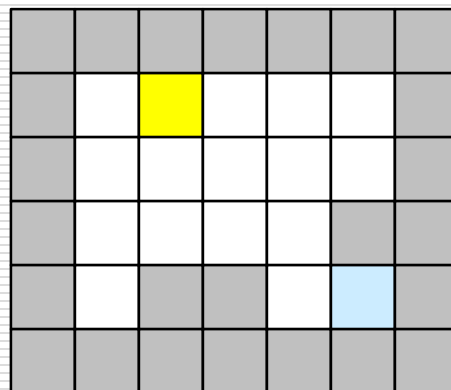
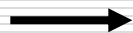
紀錄距離

$$P = 16, d = 5$$

queue 17

Input file example ↵

```
7 ↵
6 ↵
2 2 2 2 2 2 2 ↵
2 0 S 0 0 0 2 ↵
2 0 0 0 0 0 2 ↵
2 0 0 0 0 2 2 ↵
2 0 2 2 0 E 2 ↵
2 2 2 2 2 2 2 ↵
↵
```

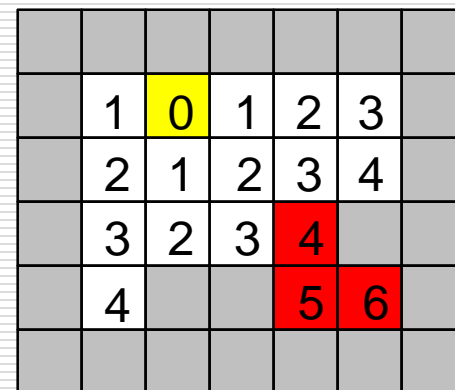


給編號

方法二其中一種解法：↵

- (1) 將起點存進queue，並記錄距離(與起點的距離)為0↵
- (2) 取出queue中的第一個"儲存的位置(P)"，且其紀錄的距離為d↵
- (3) 將P周圍"尚未存進過queue裡面的"位置存到queue裡面，並記錄距離為d+1↵
- (4) 重複(2)、(3)直到找到終點↵

從起點到終點最少需要的步數，為終點所記錄的距離(例如:終點與起點有k步的距離)；而起點到終點的路徑則是，從終點往回找 -> 距離為(k-1)的位置 -> 距離為(k-2)的位置 -> -> 距離為1的位置 -> 起點。↵



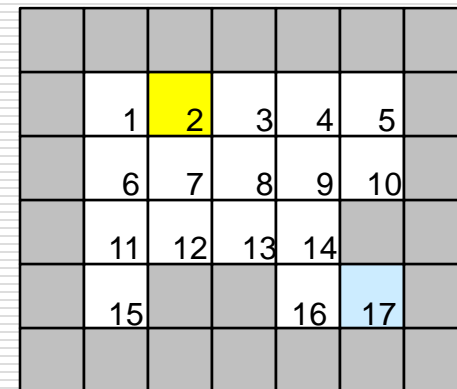
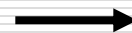
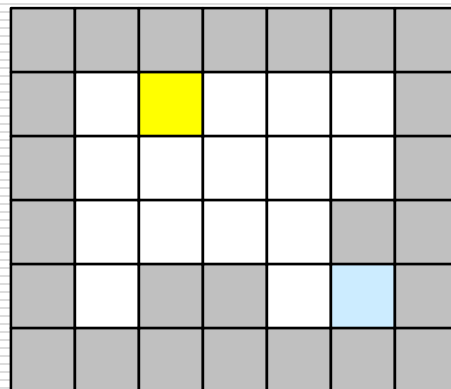
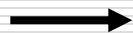
紀錄距離

$$P = 16, d = 5$$

queue 17

Input file example ↴

```
7 ↴
6 ↴
2 2 2 2 2 2 2 ↴
2 0 S 0 0 0 2 ↴
2 0 0 0 0 0 2 ↴
2 0 0 0 0 2 2 ↴
2 0 2 2 0 E 2 ↴
2 2 2 2 2 2 2 ↴
↴
```

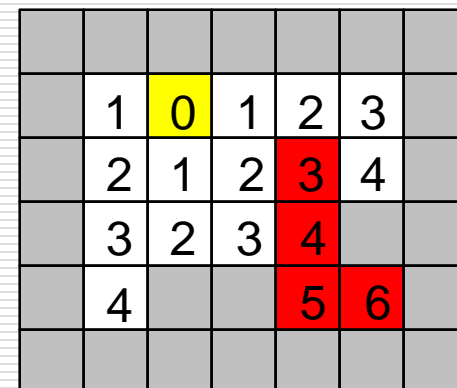


給編號

方法二其中一種解法：↴

- (1) 將起點存進queue，並記錄距離(與起點的距離)為0↴
- (2) 取出queue中的第一個"儲存的位置(P)"，且其紀錄的距離為d↴
- (3) 將P周圍"尚未存進過queue裡面的"位置存到queue裡面，並記錄距離為d+1↴
- (4) 重複(2)、(3)直到找到終點↴

從起點到終點最少需要的步數，為終點所記錄的距離(例如:終點與起點有k步的距離)；而起點到終點的路徑則是，從終點往回找 -> 距離為(k-1)的位置 -> 距離為(k-2)的位置 -> -> 距離為1的位置 -> 起點。↴



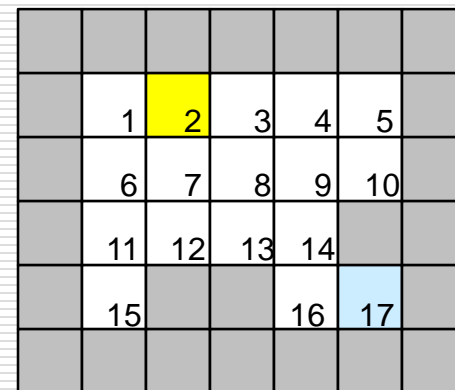
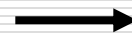
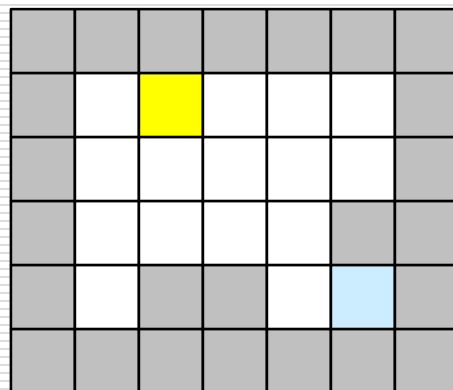
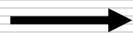
紀錄距離

$$P = 16, d = 5$$

queue 17

Input file example ↴

```
7 ↴
6 ↴
2 2 2 2 2 2 2 ↴
2 0 S 0 0 0 2 ↴
2 0 0 0 0 0 2 ↴
2 0 0 0 0 2 2 ↴
2 0 2 2 0 E 2 ↴
2 2 2 2 2 2 2 ↴
↴
```

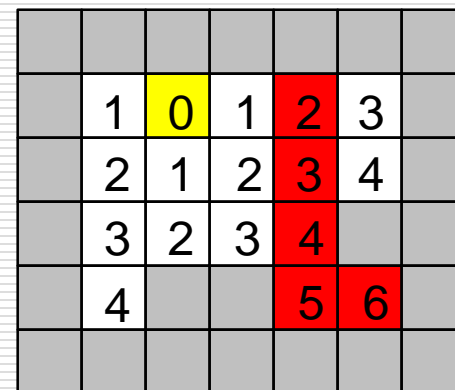


給編號

方法二其中一種解法：↴

- (1) 將起點存進queue，並記錄距離(與起點的距離)為0↴
- (2) 取出queue中的第一個"儲存的位置(P)"，且其紀錄的距離為d↴
- (3) 將P周圍"尚未存進過queue裡面的"位置存到queue裡面，並記錄距離為d+1↴
- (4) 重複(2)、(3)直到找到終點↴

從起點到終點最少需要的步數，為終點所記錄的距離(例如:終點與起點有k步的距離)；而起點到終點的路徑則是，從終點往回找 -> 距離為(k-1)的位置 -> 距離為(k-2)的位置 -> -> 距離為1的位置 -> 起點。↴



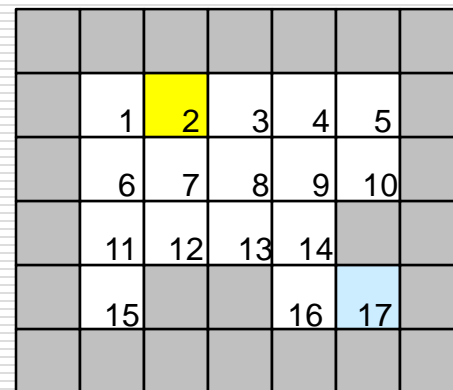
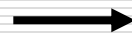
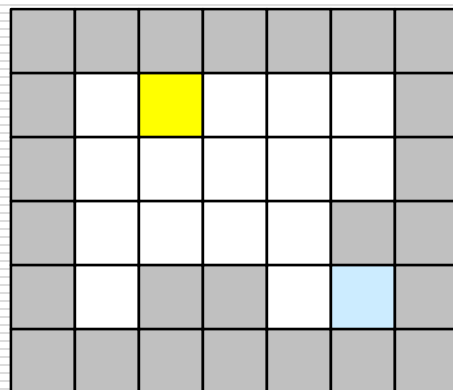
紀錄距離

$$P = 16, d = 5$$

queue 17

Input file example ↴

```
7 ↴
6 ↴
2 2 2 2 2 2 2 ↴
2 0 S 0 0 0 2 ↴
2 0 0 0 0 0 2 ↴
2 0 0 0 0 2 2 ↴
2 0 2 2 0 E 2 ↴
2 2 2 2 2 2 2 ↴
↴
```

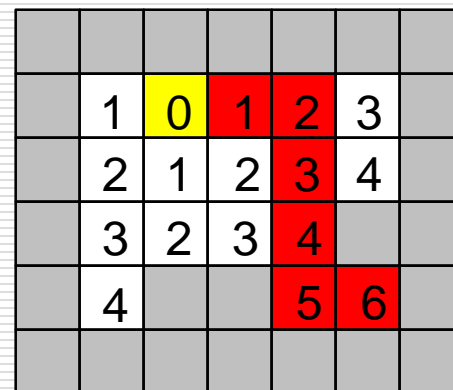


給編號

方法二其中一種解法：↴

- (1) 將起點存進queue，並記錄距離(與起點的距離)為0↴
- (2) 取出queue中的第一個"儲存的位置(P)"，且其紀錄的距離為d↴
- (3) 將P周圍"尚未存進過queue裡面的"位置存到queue裡面，並記錄距離為d+1↴
- (4) 重複(2)、(3)直到找到終點↴

從起點到終點最少需要的步數，為終點所記錄的距離(例如：終點與起點有k步的距離)；而起點到終點的路徑則是，從終點往回找 -> 距離為(k-1)的位置 -> 距離為(k-2)的位置 -> -> 距離為1的位置 -> 起點。↴



紀錄距離

$$P = 16, d = 5$$

queue 17

Thanks for your attention