

C 프로그래밍 과제#2

201912070 박승민

Num_encounter 메서드를 작성해보았습니다. Num_encounter 메서드는 Queens가 겹치는 줄이 수를 세는 메서드로 설계하였습니다.

우선, Q의 좌표를 모두 받아야겠다고 생각하여 x, y 배열을 선언하고 이 배열들에 Q의 위치 값을 넣었습니다. (Ex) 첫번째 Q의 위치가 1,2(x,y)이면, x[0] == 1, y[0] == 2)

```
53     int x[N] = { 0 };
54     int y[N] = { 0 };
55     int count = 0;
56     for (int i = 0; i < N; i++) {
57         for (int j = 0; j < N; j++) {
58             if (board[i][j] == 'Q') {
59                 x[count] = i;
60                 y[count] = j;
61                 count++;
62             }
63         }
64     }
65 }
```

X와 Y 값을 배열로 받은 후에, 겹치는 경우를 어떻게 세어야 할지 생각해보았습니다.

4가지 경우를 생각해 내었고, x, y 값을 통해서 어떻게 구해내야 할지도 생각해 보았습니다

1. 가로로 겹치는 경우 -> x값이 같은지 비교
2. 세로로 겹치는 경우 -> y값이 같은지 비교
3. 오른쪽 아래로 향하는 대각선 -> x-y(혹은 y-x) 값이 같은지 비교
4. 오른쪽 위로 향하는 대각선 -> x+y 값이 같은지 비교

우선 1, 2번에 속하는 항목들은 다음과 같이 한번에 구현하였습니다.

x_count 배열은 x의 값이 나올 수 있는 경우인 0,1,2,3,4,5,6,7 중에서 각각 개수가 몇 개인지 저장하는 배열로 정의하였습니다. 이를 for문 안에서 x의 값을 x_count 배열에 분류하여 x_count 배열의 값이 2 이상인 개수를 세었습니다. y_count 배열도 같은 방법으로 구현하였습니다.

```
66     int x_count[N] = { 0 }; // x의값 012345 가 몇개인지 경우의 수 1) 가로로 겹치는 것
67     int y_count[N] = { 0 }; // y의값 012345 가 몇개인지 경우의 수 2) 세로로 겹치는 것
68     int xcount = 0; // 겹치는 x의 값이 몇개인지
69     int ycount = 0; // 겹치는 y의 값이 몇개인지
70     for (int i = 0; i < N; i++) {
71         x_count[x[i]]++;
72         y_count[y[i]]++;
73         if (x_count[x[i]] == 2) {
74             xcount++;
75         }
76         if (y_count[y[i]] == 2) {
77             ycount++;
78         }
79     }
```

경우의 수 3, 4번은 다음과 같이 구현하였습니다. $x-y$ 값과 $x+y$ 의 값을 비교해 같은 값의 개수를 찾아야 된다는 아이디어를 도출하였고, 이를 위해 두 수의 차를 저장하는 배열인 minusarray 배열, 두 수의 합을 저장하는 배열인 plusarray 배열을 이용하여 값을 저장하고, 이중 for 문에서 같은 값을 찾으면 이를 count를 하고 하나의 값을 랜덤 값으로 바꾸어 count할 때 중복이 일어나지 않도록 하였습니다. 그리고 마지막으로 return 값은 지금까지 세었던 값을 저장한 변수들인 xcount + ycount + rightdown + rightup 변수들을 return 하여 단계별로 세었던 것들을 합쳐주었습니다.

```
81      int rightdown = 0, rightup = 0; //대각선 오른쪽 위로, 오른쪽 아래로 새는 함수
82      int minusarray[N] = { 0 }; // 경우의 수 3) 대각선 오른쪽 아래로, 이 경우,  $x-y$  값이 같은 것을 찾으면 됨
83      int plusarray[N] = { 0 }; // 경우의 수 4) 대각선 오른쪽 위로, 이 경우  $x+y$  값이 같은 것을 찾으면 됨
84      for (int i = 0; i < N; i++) {
85          minusarray[i] = x[i] - y[i];
86          plusarray[i] = x[i] + y[i];
87      }
88      for (int i = 0; i < N - 1; i++) {
89          for (int j = i + 1; j < N; j++) {
90              if (minusarray[i] == minusarray[j]) {
91                  rightdown++;
92                  minusarray[j] = rand() % 9999; // 중복값 제거를 위해 카운트 후, 랜덤 값으로 값을 날려버림
93                  break; // 중복값 제거를 위한 break;
94              }
95          }
96      }
97      for (int i = 0; i < N - 1; i++) {
98          for (int j = i + 1; j < N; j++) {
99              if (plusarray[i] == plusarray[j]) {
100                  rightup++;
101                  plusarray[j] = rand() % 9952; // 중복값 제거를 위해 카운트 후, 랜덤 값으로 값을 날려버림
102                  break; // 중복값 제거를 위한 break;
103              }
104          }
105      }
106      return xcount+ycount+rightdown+rightup;
107  }
```