

```
public static void Print(boolean[][] board, int tq, int row) {  
    if (tq == 0) {  
        display(board);  
        return;  
    }  
    for (int col = 0; col < board[0].length; col++) {  
        if (issafe(board, row, col)) {  
            board[row][col] = true;  
            Print(board, tq - 1, row + 1);  
            board[row][col] = false;  
        }  
    }  
}
```

BTFF  
PPFT  
TFPP  
FTTF

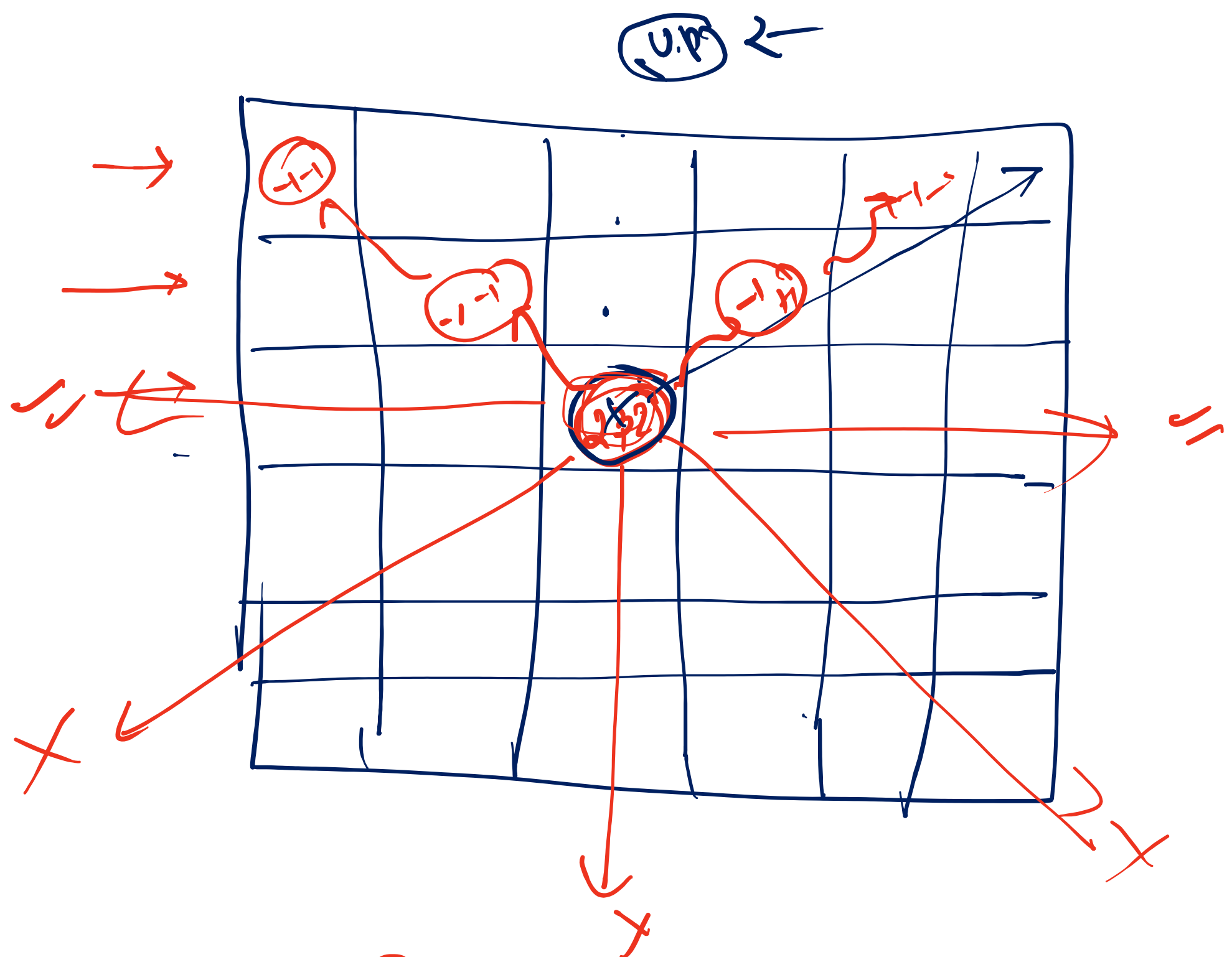
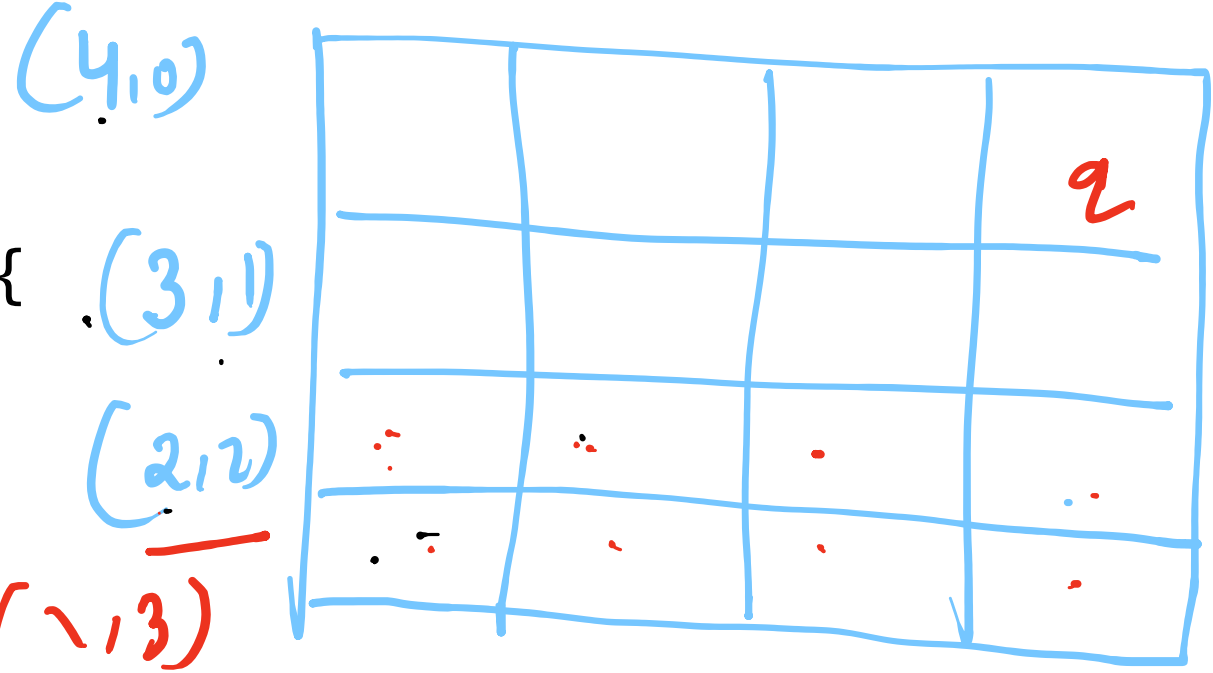
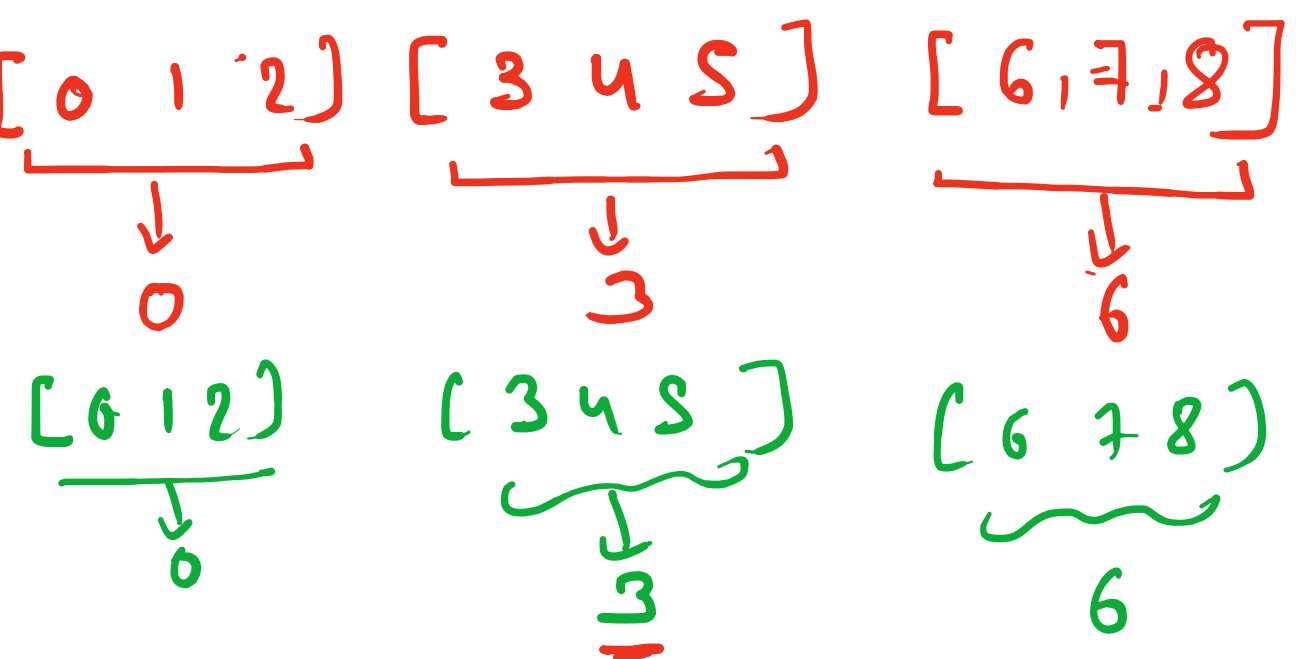


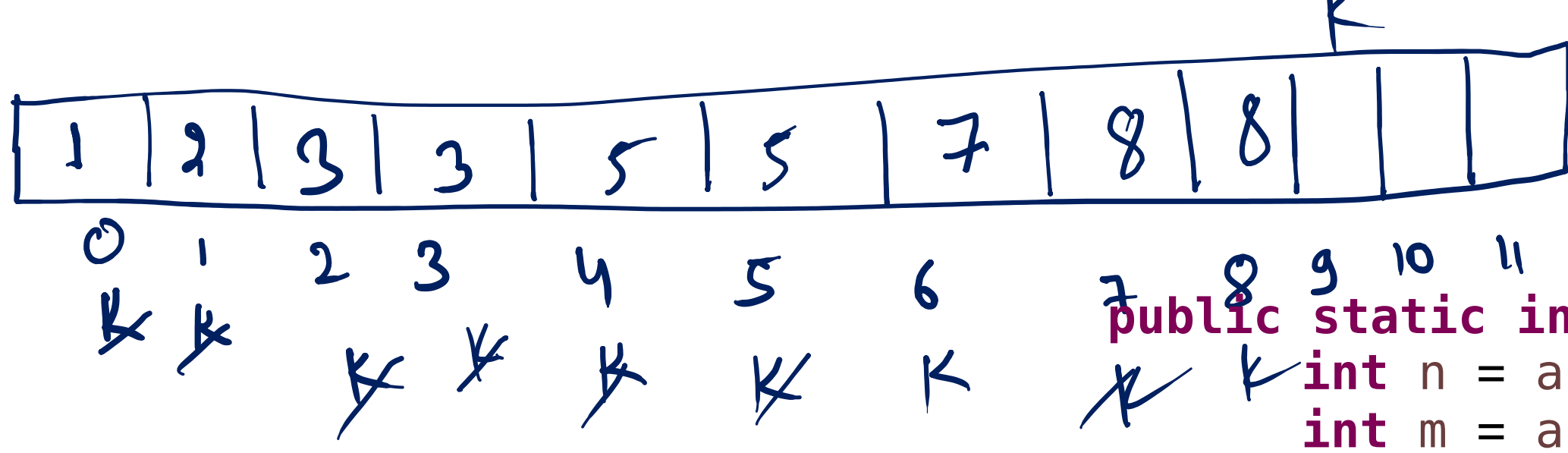
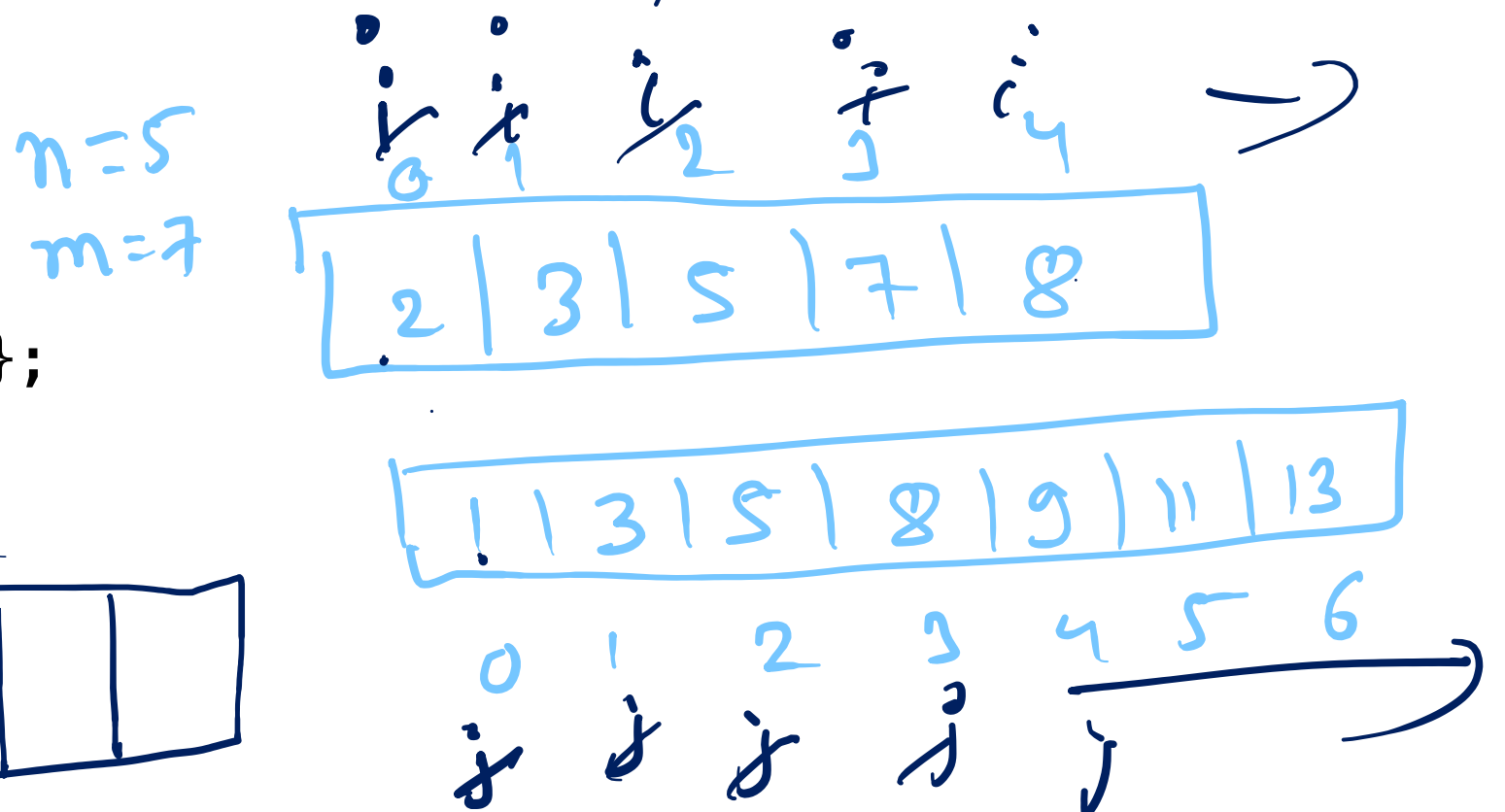
Diagram illustrating the concept of a chessboard for the N-Queens problem. It shows a grid of size  $n \times n$  with a queen piece placed on a square. The grid is labeled with  $n=3$  and  $n=4$  for different board sizes. The word "chess" is written twice, indicating the context of the problem.

```
public static void Print(int[][] grid, int row, int col) {  
    if (grid[row][col] != 0) {  
        Print(grid, row, col + 1);  
    } else {  
        for (int val = 1; val <= 9; val++) {  
            if (issafe(grid, row, col, val)) {  
                grid[row][col] = val;  
                Print(grid, row, col + 1);  
                grid[row][col] = 0;  
            }  
        }  
    }  
}
```



$(n+m) \log(n+m)$

```
int[] arr1 = { 2, 3, 5, 7, 8 };  
int[] arr2 = { 1, 3, 5, 8, 9, 11, 13 };
```



```
public static int[] Merge(int[] arr1, int[] arr2) {  
    int n = arr1.length;  
    int m = arr2.length;  
    int[] ans = new int[n + m];  
    int i = 0, j = 0, k = 0;  
    while (i < n && j < m) {  
        if (arr1[i] < arr2[j]) {  
            ans[k] = arr1[i];  
            i++;  
            k++;  
        } else {  
            ans[k] = arr2[j];  
            j++;  
            k++;  
        }  
    }  
}
```

Left 4 Right

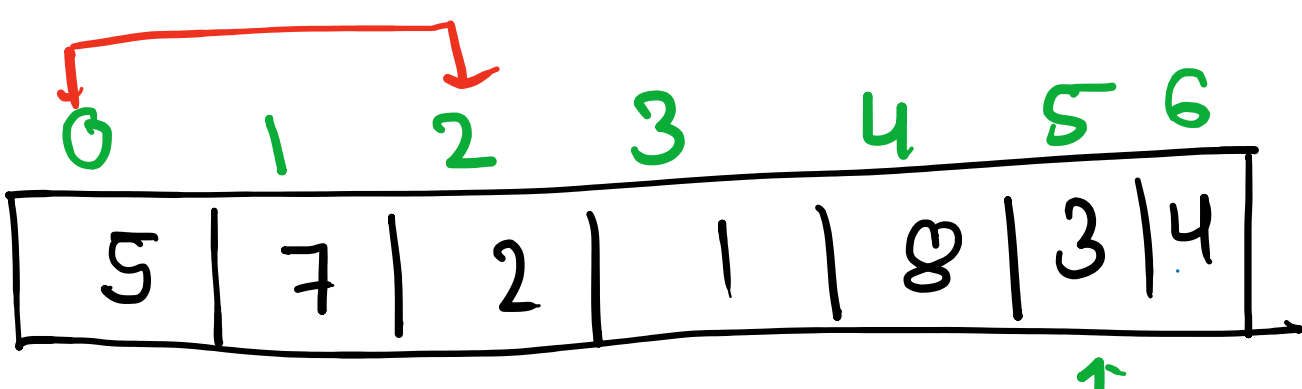


Diagram illustrating the concept of a chessboard for the N-Queens problem. It shows a grid of size  $n \times n$  with a queen piece placed on a square. The grid is labeled with  $n=3$  and  $n=4$  for different board sizes. The word "chess" is written twice, indicating the context of the problem.

3 swap (idx, idx)