

N-Queens Problem

Public class NQueens {

Public static void solveNQueens (int n) {

int[] placement = new int[n];

placeQueens (placement, 0, n);

}

Public static placeQueens (int[] placement, int row, int n) {

• if (row == n) {

PrintSolution (placement);

return;

}

for (int col = 0; col < n; col++) {

if (isValid (placement, row, col)) {

placement [row] = col;

placeQueens (placement, row+1, n);

}

}

}

Public static boolean isValid (int[] placement, int row, int col) {

for (int i = 0; i < row; i++) {

if (placement[i] == col || Math.abs (placement[i] - col)

== Math.abs (i - row)) {

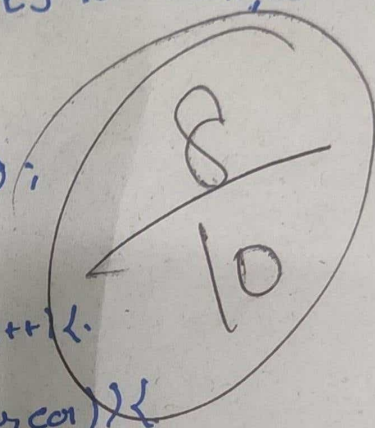
return false;

}

return true;

Public static void

PrintSolution (int[] placement) {



for (int col : placement) {

for (int i = 0; i < placement.length; i++) {

if (i == col) {

System.out.println("Q");

System.out.print(" ");

System.out.print(" ");

System.out.print(" ");

System.out.print(" ");

System.out.print(" ");

System.out.print(" ");

System.out.print(" ");

System.out.print(" ");

System.out.print(" ");

System.out.print(" ");

System.out.print(" ");

System.out.print(" ");

System.out.print(" ");

System.out.print(" ");

System.out.print(" ");

System.out.print(" ");

System.out.print(" ");

System.out.print(" ");

Selection Sort:

```
public static void selection(int arr[]) {
```

```
    int n = arr.length;
```

```
    for (int i = 0; i < n; i++) {
```

```
        int max = arr[i];
```

```
        for (int j = i + 1; j < n; j++) {
```

```
            if (arr[j] > max) {
```

```
                max = arr[j];
```

```
            int temp = arr[i];
```

```
            arr[i] = max;
```

```
            max = arr[j];
```

```
        }
```

```
    }
```

```
public static void main (String[] args) {
```

```
    int[] arr = { 2, 3, 9, 7, 5, 8 };
```

```
    selection(arr);
```

```
}
```