boundarySort.java

```java
// Sort only boundary elements of a 2d array

import java.util.Arrays;
import java.util.Scanner;

public class boundarySort {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter number of rows: ");
        int r = sc.nextInt();
        System.out.println("Enter number of coln: ");
        int c = sc.nextInt();
        System.out.println("Enter " + (r * c) + " elements: ");
        int[][] arr = new int[r][c];
        // Store boundary elements of arr in a new 1d array
        int[] boundary = new int[(r * c) - (r - 2) * (c - 2)];
        int count = 0;
        for (int i = 0; i < r; i++) {
            for (int j = 0; j < c; j++) {
                arr[i][j] = sc.nextInt();
                if (i == 0 || j == 0 || i == (r - 1) || j == (c - 1)) {
                    boundary[count] = arr[i][j];
                    ++count;
                }
            }
        }
        sc.close();
        // Sort the 1d array using selection sort
        int min, temp;
        for (int i = 0; i < boundary.length; i++) {
            min = i;
            for (int j = i + 1; j < boundary.length; j++) {
                if (boundary[j] < boundary[min])
                    min = j;
            }
            temp = boundary[i];
            boundary[i] = boundary[min];
            boundary[min] = temp;
        }
        // Put the sorted elements back in 2d array
        count = 0;
        for (int i = 0; i < c; i++) {
            arr[0][i] = boundary[count];
            ++count;
        }
        for (int i = 1; i < r; i++) {
            arr[i][c - 1] = boundary[count];
            ++count;
        }
        for (int i = c - 2; i >= 0; i--) {
            arr[r - 1][i] = boundary[count];
            ++count;
        }
        for (int i = r - 2; i >= 1; i--) {
            arr[i][0] = boundary[count];
            ++count;
        }
        for (int i = 0; i < r; i++) {
            System.out.println(Arrays.toString(arr[i]));
        }
    }
}
/**
 * Variable        Data        Table
 * r, c            int         Store rows and columns for 2d array
 * arr             int[][]     Store 2D array
 * boundary        int[]       Extract and store boundary elements in 1d array
 * count, i, j     int         Iterators
 * min, temp       int         Used in sorting
 */
```