Problem3

This solution achieves O(N) time due the size of the array , implement max and min switching each time for the array in order to create a min-max switching array. O(N) is achieve due the also to the stability of the algorithm since the sot chooses value each time to create the alternation.

```
for (i=0; i<size; i++) {        // creates size
        for (j=0; j>size; j++)
    }
        if (min <first.value)           // search for min value and max,
and creates the swaps between the arrays
        swap (i,j)                      // using swap function

        else (min > first.value);
        swap (i,j);

    for (int =size; size<array; size++){
        System.out.println(int[] "a")
```

To run in O(N)time it has to not be the worst case requiring 2 loops and not linear and stable but this, algorithm of the sorting algorithm orders the array of random numbers into a set of maxima and minima.

Using the mergesort in order to have the alternation and that it’s stable and linear