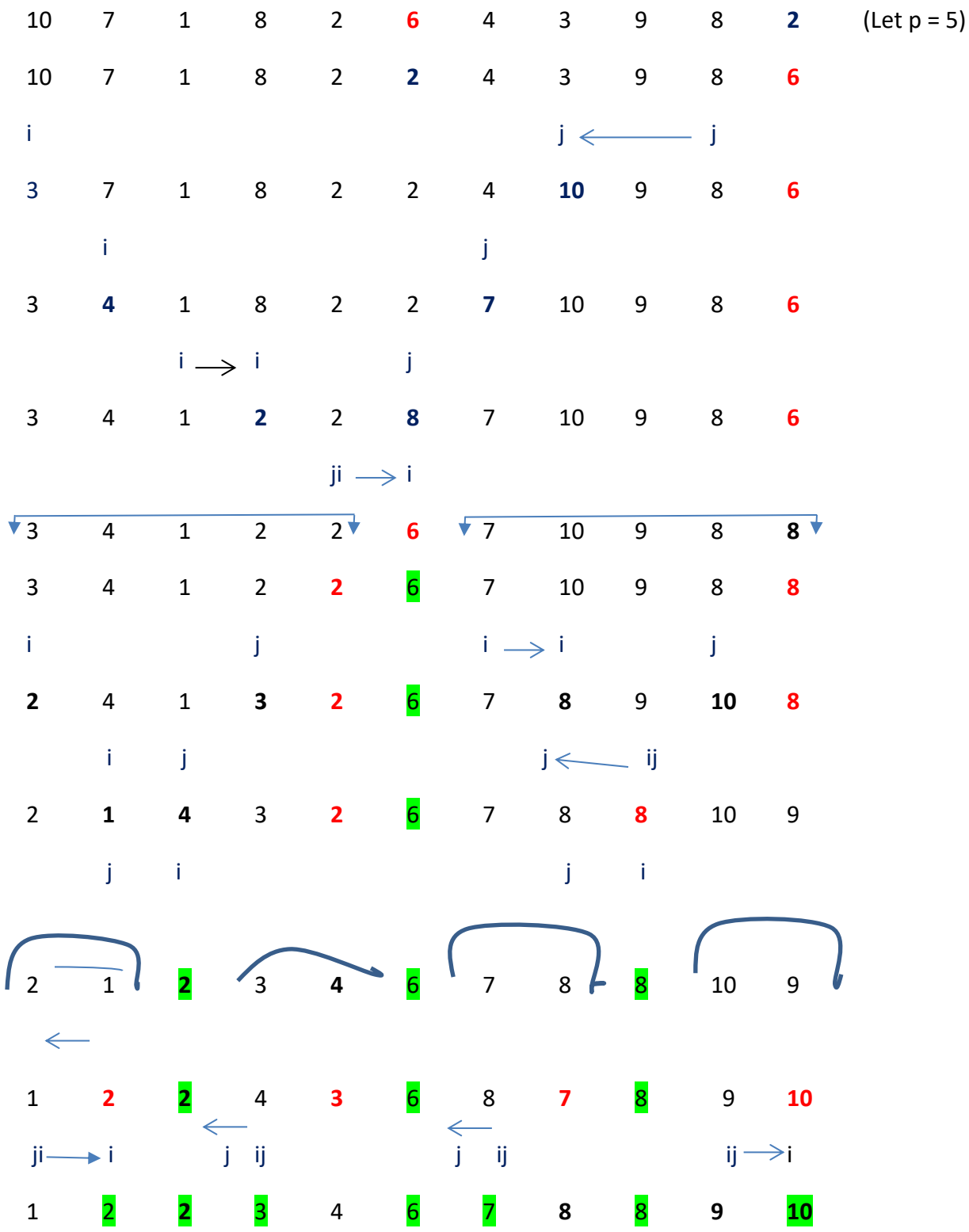
Algorithm Quicksort

Major steps are as follows:

1. Pick a pivot (We will always pick pivot as median of three in our course)
2. Swap pivot with the last item in the array.
3. Place i at the first item of the array.
4. Place j at the last - 1 item of the array.
5. So long as i less than or equal to j do the following:
    1. Move i so that i is pointing to a value greater than or equal to the pivot.
    2. Move j so that j is pointing to a value less than or equal to the pivot.
    3. Then swap values pointed by i and j.
    4. Move i one step.
    5. Move j one step.
6. After the crossover, swap values pointed by i and the pivot.
7. Now you have divided array into three: L, pivot, R.
8. Now recursively call L and recursively call R.

```
private void quicksort(int[] arr, int start, int stop) {
            int i, j;        // these are the moving pointers

            if (stop <= start) return;
            else {
                  int pivotPos = medianOfThree(arr, start, stop);
                  swap(arr, pivotPos, stop);
                  int pivot = arr[stop];
                  i = start;
                  j = stop - 1;
                  while (true) {
                        while (i<=j && arr[i] < pivot) i++;
                        while (i<=j && arr[j] > pivot) j--;
                        if(i <= j) {
                              swap(arr, i++,j--);
                        }
                        else break;
                  }
                  swap(arr, stop, i);
                  quicksort(arr, start, i-1);
                  quicksort(arr, i+1, stop);
            }
      }
```


10 7 1 8 2 6 4 3 9 8 2 (Let p = 5)
10 7 1 8 2 2 4 3 9 8 6
i
j
j
3 7 1 8 2 2 4 10 9 8 6
i
j
3 4 1 8 2 2 7 10 9 8 6
i
i
j
3 4 1 2 2 8 7 10 9 8 6
ji
i
3 4 1 2 2 6 7 10 9 8 8
3 4 1 2 2 6 7 10 9 8 8
i
j
i
i
j
2 4 1 3 2 6 7 8 9 10 8
i
j
j
ij
2 1 4 3 2 6 7 8 8 10 9
j
i
j
i
2 1 2 3 4 6 7 8 8 10 9
1 2 2 4 3 6 8 7 8 9 10
ji
i
j
ij
j
ij
ij
i
1 2 2 3 4 6 7 8 8 9 10