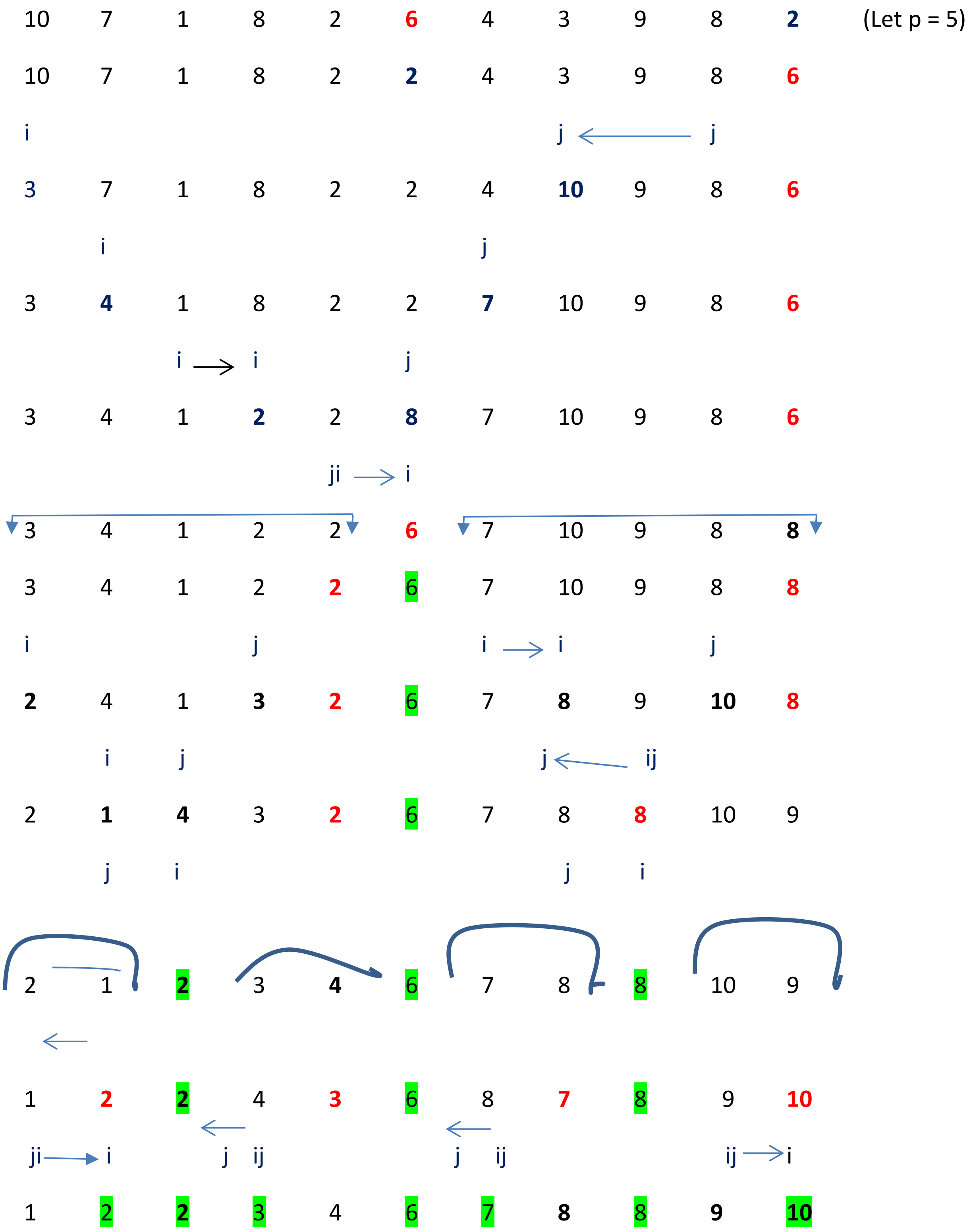
Algorithm Quicksort

```
private void qsort(int[] keyArray, int lower, int upper) {
        int lowPtr, highPtr;            // these are the moving pointers into the array

        if (upper <= lower) return;
        else {
            int pivotPos = medianOfThree(keyArray,lower,upper);
            swap(keyArray,pivotPos,upper);
            int pivot = keyArray[upper];
            lowPtr = lower;
            highPtr = upper-1;
            while (true) {
                while (lowPtr<=highPtr && keyArray[lowPtr] < pivot) lowPtr++;
                while (lowPtr<=highPtr && keyArray[highPtr] > pivot) highPtr--;
                if(lowPtr <= highPtr) {
                    swap(keyArray, lowPtr++,highPtr--);
                }
                else break;
            }
            swap(keyArray, upper,lowPtr);
            qsort(keyArray,lower, lowPtr-1);
            qsort(keyArray,lowPtr+1, upper);
        }
    }
```



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	