Find The Median

```
MedianFind(A[0..n-1])
     Let A[p] be the pivot //TODO need to select p.
     Let S and B be two arrays //"small" and "big" elements
     for(i from 0 to n-1 except p)
          if(A[i] <= A[p])
               Copy A[i] into S
          else
               Copy A[i] into B
     if(S.length == n/2) return A[p] //A[p] is median
     else if (S.length > n/2)
          …//TODO what goes here?
     else
          …//TODO what goes here?
```

Examples

6	3	1	8	10	2	5	11	9	7	4

$n = 11$, median is 6

Pivot $A[0] = 6$

3	1	2	5	4	**6**	8	10	11	9	7

Pivot $A[10] = 4$

3	1	2	**4**	6	8	10	5	11	9	7

Pivot $A[3] = 8$

6	3	1	2	4	7	4	**8**	10	11	9

Pivot Finding

You can still find the median of 5 elements in constant time. (5 is a constant).

Don't just find the median of 5 elements and make that a pivot

…split the array into groups of 5 and get $n/5$ candidate pivots

If the array starts

32	5	17	53	101	10	4	23	15	98	…

Blue medians are candidates

5	4														
17	10														
32	15														
53	23														
101	98														

Selection

```
QuickSelect(A[0..n-1], k)
     p = PivotFinder(A)
     Let S and B be two arrays //"small" and "big" elements
     for(i from 0 to n-1 except A[p])
          if(A[i] <= A[p])
               Copy A[i] into S
          else
               Copy A[i] into B
     if(S.length == k - 1) return A[p] //A[p] is index k
     else if (S.length > k -1)
          QuickSelect(S, k)
     else
          QuickSelect(B, k-S.length)
```

$O(n)$ and a recursive call