Finding the k-th smallest integer

The most obvious one is by sorting the array and picking the k-th element of the array. But this method does more than what it is requested. There is another method, which will be described here, that picks the k-th smallest integer.

Problem Description

We have an arbitrary array of integer [first..last]. We have to choose the k-th smallest integer.

Algorithm

1. Choose a pivot from the array
2. Partition the array so that : A[first...pivotIndex-1] <= pivot <= A[pivotIndex+1..last]
3. if k < pivotIndex then it must be on the left of pivot, so do the same method recursively on the left part
4. if k = pivotIndex then it must be the pivot
5. if k > pivotIndex then it must be on the right of pivot, so do the same method recursively on the right part

Implementation in C++

```
/* Problem : Finding the k-th smallest value
 * Author  : Stephanus
 * Lang.   : C++
 * Date    : 19 January 2004
 */

#include < iostream >

using namespace std;

// return a pivot index and array with
// array[first..pivot-1] <= array[pivot] <= array[pivot+1..last]
int partition(int *array,int first,int last)
{
        int pivot = array[first];
        int i = first-1;
        int j = last+1;

        while (1)
        {
                do { j--; } while (array[j] > pivot);
                do { i++; } while (array[i] < pivot);
                if (i < j)
                {
                        array[i] ^= array[j];
                        array[j] ^= array[i];
                        array[i] ^= array[j];
                } else
                        return j;
        }
}
```

```
int ksmall(int *array,int k,int first,int last)
{
        int pivotIndex = partition(array,first,last);

        if (k < pivotIndex - first + 1)
                return ksmall(array,k,first,pivotIndex-1);
        else if (k == pivotIndex - first + 1)
                return array[pivotIndex];
        else
                return ksmall(array,k - pivotIndex + first -
1,pivotIndex+1,last);
}

int main()
{
        int tmp[10] = { 3, 4 , 6, 2, 4, 6,0, 4, 1,324 };

        cout << ksmall(tmp,1,0,9) << endl;
        return 0;
}
```