

```

class Solution {

    public int partition(int arr[],int s,int e)

    {

        int pivot=arr[s];

        int i=s,j=e;

        while(i<j) {

            do{

                i++;

            } while(arr[i]<pivot);

            do {

                j--;

            }while(arr[j]>pivot);

            if(i<j){

                int t=arr[i];

                arr[i]=arr[j];

                arr[j]=t;

            }

        }

        int t=arr[s];

        arr[s]=arr[j];

        arr[j]=t;

        return j;

    }

    public int quicksort(int arr[],int s,int e,int k){

        if(s<e){

            int j=partition(arr,s,e);

            if(k<j)quicksort(arr,s,j,k);

            else if(k>j)quicksort(arr,j+1,e,k);

            else{

                return arr[j];

            }

        }

    }

```

```
    }

    return arr[s];

}

public int findKthLargest(int[] nums, int k) {

    int n=nums.length;

    int arr[]=new int[n+1];

    for(int i=0;i<n;i++) {

        arr[i]=nums[i];

    }

    arr[n]=Integer.MAX_VALUE;

    quicksort(arr,0,n,n-k);

    return arr[n-k];

}

}
```