**National Institute of Technology, Calicut**

**Department of Computer Science and Engineering**

CS2094 - Data Structures Lab

**Assignment #2 (Advanced)**

1 You have to find the kth smallest element in a given list of strings, in the most efficient manner possible.

Input: An integer k, and a filename <filename> to a file containing an unknown number of strings. The strings have a maximum length of 6, are composed only of lower case alphabets and are separated by spaces.

Output: The kth smallest element in <filename> according to lexicographic ordering

Your program should terminate with appropriate error messages in case of invalid input.

The program should use O(1) space, and a time complexity as small as you can make it.

(Hint: Remember that sometimes K could be O(n), for example n/2. Sometimes k could be a small constant. Your algorithms should be efficient under all circumstances.

Write your program, and measure its running time, on files of sizes 10 to 100000.

2. Consider how you would sort a file that was too large to fit in your machines RAM. External Sorting refers to the sorting of data which is not in the primary memory of the computer. In this problem, you have to implement a miniature version of external sort. The filesizes have been intentionally kept small because of time constraints. The methodology followed has to be the same as for really big files.

a. First generate a 256 MB file Bigfilecontaining integers. You can make use of a random number generator, with suitable modifications on the generator output, to generate suitable integers.

b. Then sort the above file in the following manner.

1. Bring in 1024\*1024 integers into memory at a time, and sort into an output file, thus generating 64 sorted files of integers.
2. Do an 8 way merge on the contents of these files to generate 8 sorted files.
3. Do a 4-way merge on the above files to generate 2 sorted files
4. Merge the two files obtained above into Bigfile, to obtain the sorted Bigfile.

3. Implement an algorithm to find the median of ***n*** integers in **O(*n*)** time in all cases and with θ(1) space complexity.(Reference: Cormen chapter9)

Input : n integers

Output: Median of the n integers.

Sample Input

7

712

-45

456765

0

-8907

4566

3456

Output

712

Sample Input

6

23

5

2

4

1

3

Output

3.5