CS2100 Computer Programming Lab
Divide and Conquer

Release Date: October 12, 2015
Due Date: October 25, 2015

Problem 1 : Majority in an Array

An array $A[1, \ldots, n]$ is said to have a majority element if more than half of its entries are equal. Our goal in this problem is the simple task of determining if an array has a majority element. However, as is usually assumed with integers the elements of the array cannot be ordered. Therefore, there can be no comparisons of the form is "$A[i] > A[j]$" However, assume that you can answer questions of the form: "is $A[i] == A[j]$" efficiently. Assume that a function isequal returns the answers to such queries.

A naive algorithm would be to count for every element $A[i]$, how many elements in the array are equal to $A[i]$. This algorithm requires n^2 queries to the isequal function. We ask you to implement an algorithm using divide and conquer which requires fewer than n^2 queries. Consider splitting the array A into two arrays A_1 and A_2. Does knowing the majority elements of A_1 and A_2 help you find the majority element of A?
Important: Implementing the naive algorithm will not fetch any credit although it gives correct output.

Input Output

Assume that each element in the array is an integer. The input is given to you in a file which has the following format. The first line is the integer n (assume that n is a power of 2) which denotes the number of integers in the array. A sample input file looks like inputfile.txt given below.

inputfile.txt

```
8
10
30
15
5
3
60
33
888
```

definitions of equality

- Two elements e_1 and e_2 are equal iff $e_1 = e_2 \mod 5$.
  10, 30, 15, 5, 60 are all equal. Any one of them can be output as majority element.
- Two elements e_1 and e_2 are equal iff $e_1 \div 10 = e_2 \div 10$.
  5 and 3 are equal; 10 and 15 are equal. The input has no majority element under this definition of equality.

In addition, assume that a function isequal is supported. The function isequal has the following signature:
int isequal (int e1, int e2): The function returns 1 if element e1 is equal to element e2; it returns 0 otherwise.
You should define your own function isequal – we will define our own while testing. Let us say for the above input we have the following definition of isequal: Two elements are equal if and only if the two integers are equal modulo 5. Under such a definition of equality the elements "10, 30, 15, 5, 60" are all equal and therefore they form a majority. Your output may be any one of these integers as majority. Output the majority element in a file. In case there exists no majority element, output "No majority element" in the file. Your program should accept the inputfile and the outputfile names as command line arguments.

Problem 2 : Order Statistics on Sorted Arrays

Recall the definition of median of a set of elements. For a sorted array containing $2k + 1$ integers say $A[1, \dots, 2k + 1]$, the median is $A[k + 1]$. For a sorted array containing $2k$ integers $A[1, \dots, 2k]$, the median can be defined as $(A[k] + A[k + 1])/2$. In this problem we are given two sorted arrays A_1 and A_2 each containing n integers. Our goal is to find the median of array A obtained after merging A_1 and A_2.

As in the previous question, there is a naive method of doing it while merging the two arrays. However, we would like to use divide and conquer to get an efficient solution. If the sizes of the two arrays are equal to 2, then use the naive algorithm and compute the median. Else, if the sizes are greater than 2, then let m_1 and m_2 be the medians of A_1 and A_2 respectively. Use m_1 and m_2 to determine how to reduce the sizes of the two arrays to approximately half the original size and invoke median on the two smaller subarrays. In particular, think about the three cases that can arise:

- $m_1 == m_2$.
- $m_1 < m_2$.
- $m_1 > m_2$.

Important: Do not search for solutions for the problem on the internet or discuss it with your friends. Try to come up with the best solution you can on your own. As in the first question, the naive algorithm that tries to merge the two arrays and maintains a count, will not fetch any credit.

Input Output

Assume that the input is given to you in a file which has 3 lines. The first line gives you n which is the number of elements in both the sorted arrays A_1 and A_2. The next two lines contain the sorted integers in A_1 and A_2 respectively. Assume that the lines are comma separated. A sample input-file is as shown below.

input-file.txt	**output-file.txt**
10	82.5
1, 7, 12, 30, 102, 105, 111, 123, 200, 210	
3, 7, 13, 22, 79, 80, 85, 203, 204, 222	

Your program must accept two files – inputfile and output file as command line arguments and store the median in the outputfile.

Submission Guidelines

Follow the usual submission guidelines and submit two different files containing the codes for Problem1 and Problem2. Remove all temporary files and object files and make a tar.gz and upload the tar ball to moodle.