

APS - Monsoon 2018

Assignment - 1

NOTE : Use of any inbuilt or third party library is not allowed. (including STLs and Collections)

Problem 1

Statement : Your task is to create a large integer library, similar to what we have in Java as BigInteger. Your library should provide functionalities to store arbitrarily large integer and perform basic math operations.

Operations to be implemented : Add, Subtract, Multiply, Divide, GCD of 2 large integers.

Languages allowed : C, C++, Java (Use of any third party or inbuilt large integer library is not allowed)

Evaluation parameters : Accuracy of operations and performance.

Submission: Hackerrank

Problem 2

Statement : Create an AVL Tree for integers without using any other third party library. Your tree should work in a way such that a duplicate can not exist in it (similar to the set implementation of C++ STL)

Operations to be implemented : Insertion, Deletion, Searching for an element, searching for Kth smallest element among all the inserted values.

Languages allowed : C, C++, Python, Java

Evaluation parameters : Worst case Log N insertion, deletion and search time for elements in the tree.

Submission: Hackerrank

Problem 3

Statement : The questions aims at the implementation of K-Way merge sort algorithm to sort very large arrays. This is specially useful when arrays are large enough and thus do not fit in the memory. This algorithm is a perfect example of the use of divide and conquer where with limited resources large problems are tackled by breaking the problem space into small computable subspaces and then operations are done on them.

Input : A file containing a large unsorted list of integers (May be large enough that they do not fit into your RAM at once)

Output : A file containing non Descending sorted list of the given integers

Languages allowed : C, C++, Python

Evaluation parameters :

1. Time and Space Used by the algorithm
2. Efficient use of Data-Structures

Submission format: Your code should take two arguments. First is the name of input file. Second is name of output file.

Eg.

If your input file is at ./data/input.txt

And if you need your output file at ./data/ named output.txt

For c++, code should be of format ROLL_NO_3.cpp

compiled file should accept two arguments

./a.out "./data/input.txt" "./data/output.txt"

For Java, code should be named ROLL_NO_3.java

Compiled file will be ROLL_NO_3

java ROLL_NO_3 "./data/input.txt" "./data/output.txt"

For python, code should be named ROLL_NO_3.py

python ROLL_NO_3.py "./data/input.txt" "./data/output.txt"

Problem 4

Statement : Implementation of vectors.

What is vector?

- Vectors are same as dynamic arrays with the ability to resize itself automatically when an element is inserted or deleted, with their storage being handled automatically by the container.
- Vector elements are placed in contiguous storage so that they can be accessed and traversed using iterators.
- In vectors, data is inserted at the end. Inserting at the end takes differential time, as sometimes there may be a need of extending the array.
- Removing the last element takes only constant time because no resizing happens.
- Inserting and erasing at the beginning or in the middle is linear in time.

What is expected as solution?

- The C++ standard specifies that a legal (i.e., standard-conforming) implementation of vector must satisfy the following performance requirements:
 - `vector()` - initialize a blank vector
 - `vector(n,x)` - initialize a vector of length n with all values as x.
 - `size()` - return current size of vector.
 - `v[i]` - return value at i^{th}
 - `push_back(x)` - append data x at the end.
 - `pop_back()` - erase data at the end.
 - `insert(i,x)` - insert value x at i^{th} index and right shift the later values.
 - `erase(i)` - erase value at i^{th} and left shift all later values.
 - `front()` - return the first element(value) in the vector.
 - `back()` - return the last element(value) in the vector.

Submission format:

Code should be of format ROLL_NO_4.cpp

Problem 5

Median of Array :

It is defined for sorted sequence . In a sorted array of integers , median of the array is defined as follows

- If there are odd numbers of elements in an array , median element is the middle of that array E.x [1 , 2 , 3] median is 2 and for [1 , 3 , 3] median is 3
- If there are even numbers of elements in an array , median element is average of two middle elements of that array E.x [1 , 2 , 3 , 4] median is $(2 + 3) / 2$ and for [1 , 3 , 3 , 5] median is 3

Problem Description :

Given a stream of integers , after occurrence of every number , output the median of array

Example : - given sequence is 5 , 7 , 1 , 6 , 2 , 3

Explanation: - [5] median is 5
[5 , 7] median is $(5+7)/2$
[5 7 1] sorted array is [1 5 7] median is 5
[5 7 1 6] sorted array is [1 5 6 7] median is $(5 + 6)/2$

Submission: Hackerrank.