

Lab 6: Introduction to Data Structures

Topics covered: Linear sorting algorithms and AVL trees

Max points: 30

Q1) Implementing Linear sorting algorithms: Write a C++ program to implement: (15)

- 1) Radix Sort
- 2) Counting Sort
- 3) Bucket Sort

Record the time it takes to perform the sort in each case. For a given input list from the end-user, output the sorted list with the time of execution.

Initial print message

Please select an algorithm to perform the sorting:

- 1) Radix sort
- 2) Counting sort
- 3) Bucket sort

Input: 1

Print message:

Please enter the number of elements and the elements to be sorted.

Input:

10

23 14 25 89 75 78 84 455 233 122

Output:

14 23 25 75 78 84 89 122 233 455

0.567 s

Hint: You may use the `clock()` function to compute the time of execution. An example is provided below from *Geeks for Geeks*:

<https://www.geeksforgeeks.org/measure-execution-time-with-high-precision-in-c-c/>

```
#include <bits/stdc++.h>
using namespace std;
```

```
// A sample function whose time taken to
// be measured
void fun()
{
```

```

    for (int i=0; i<10; i++)
    {
    }
}

int main()
{
    /* clock_t clock(void) returns the number of clock ticks
       elapsed since the program was launched.To get the number
       of seconds used by the CPU, you will need to divide by
       CLOCKS_PER_SEC.where CLOCKS_PER_SEC is 1000000 on typical
       32-bit system. */
    clock_t start, end;

    /* Recording the starting clock tick.*/
    start = clock();

    fun();

    // Recording the end clock tick.
    end = clock();

    // Calculating total time taken by the program.
    double time_taken = double(end - start) / double(CLOCKS_PER_SEC);
    cout << "Time taken by program is : " << fixed
         << time_taken << setprecision(5);
    cout << " sec " << endl;
    return 0;
}

```

Q2) Write a C++ program to implement the median of medians algorithm to find the jth max and compare the effect on the time-of-execution with the increase in the number of inputs: (7 marks)

- 1) Computing jth max by naive algorithm i.e. computing 1st min, then 2nd min, .., and so on
- 2) Computing jth max by random pivot algorithm (using partition algorithm by quick sort)
- 3) Computing jth max by the median of medians algorithm

You need to construct two plots for each of the cases above:

- 1) fix j to some value (say n/10) and vary n from 100, 200, 300, 400, 500
- 2) fix the value of n (say 500) and vary j from 1, 10, 50, 100, 150, 200, 250

Q3) Write a C++ program to perform AVL tree insertion and deletion. For a given end-user input, perform the insertion and deletions and show the tree structure before and after the deletion as a level order traversal. (8 marks)

Print message: "Select the operation and enter the number to inserted or deleted.

- 1) Insertion
- 2) Deletion"

Input:

1 5

Output:

List before the operation: Null

List after the operation: 5

Input:

2 6

List before the operation: 5

List after the operation: 5

6 is not present in the tree