Birla Institute of Technology and Science, Pilani Hyd Campus
CS F211: Data Structures and Algorithms

2nd Semester 2024-25 Lab No:5

Overview:

The goal of this lab is to learn how to implement **recursive functions** to perform operations on data structures and to compute **run-time complexity** of a program through **empirical means** (by using **Chrono** library in C++ that provides a set of functions to work with time).

Program 1: In the previous lab, you learnt about few basic operations on singly linked list and doubly linked list. For this lab, we have changed few of those basic functionalities to be recursive like insert (), delete (), print ().

```
+---------------------------------------+
Please enter one of the following choices:
1 : Insert at end
2 : Delete from end
3 : Print Forward
4 : Print Backward
5 : Reverse List
6 : Get N'th node from the end
7 : Exit
1
Enter the item to insert: 10
Time spent: 0.002 ms.
+---------------------------------------+
Please enter one of the following choices:
1 : Insert at end
2 : Delete from end
3 : Print Forward
4 : Print Backward
5 : Reverse List
6 : Get N'th node from the end
7 : Exit
3
10
Time spent: 0.024 ms.
+---------------------------------------+
```

```
+---------------------------------------+
Please enter one of the following choices:
1 : Insert at end
2 : Delete from end
3 : Print Forward
4 : Print Backward
5 : Reverse List
6 : Get N'th node from the end
7 : Exit
1
Enter the item to insert: 20
Time spent: 0.003 ms.
+---------------------------------------+
Please enter one of the following choices:
1 : Insert at end
2 : Delete from end
3 : Print Forward
4 : Print Backward
5 : Reverse List
6 : Get N'th node from the end
7 : Exit
3
10 20
Time spent: 0.011 ms.
+---------------------------------------+
```

Few functions in the given file (**Program1.cpp**) are incomplete. Pl. complete those and run the code to get the above output. You may think to include the header file (**timer.h**) given along with this document. Incomplete functions are as given below:

```
template <typename DT>
void SinglyLinkedList<DT>::printForward()
{
   //missing code
}
```

```
template <typename DT>
void SinglyLinkedList<DT>::printBackward()
{
   // missing code
}
```

Task:

Write down the missing code in the above two functions (printForward, and printBackward). Analyse why the time spent on carrying out different operations are different, and if there is any correlation between these times. In the given timer.h, Chrono library is using a high-resolution clock, a clock with the shortest tick period available.

(**Note:** Write your observations in a word file and upload it to the lab course page at the end of the lab).

Program 2: Factorial of a non-negative number n is the product of all positive integers less than or equal to n. The given file (Program2.cpp) has 3 different functions to calculate the factorial of a non-negative number. The functions use non-recursive, tail-recursive and non-tail recursive approaches to calculate factorial of a number. Understand the working of these functions.

Next, you should use a GNU profiling tool named gprof as discussed in the lecture class**.** Through profiling one can determine the parts in program code that are time consuming and need to be re-written. This helps make your program execution faster which is always desired.

You should use the tool to observe the time taken by the different functions to execute on a machine.

Execute the following commands on the bash prompt to use the profiler:

```
$g++ Program2.cpp -o Prog2 -pg
$./Prog2
$gprof Prog2 > Prog2.txt
```

A file named Prog2.txt will be generated. Observe the time taken by the different functions. Note the differences between time taken by a recursive and an iterative (non-recursive) function. Also, note the difference between a tail recursive and a non-tail recursive function.
(May read more at: http://web.cecs.pdx.edu/~karavan/perf/book_gprof.html)

(**Note:** Take different values of 'n' and run it multiple times, and plot a graph of input size and time taken (observed using gprof). Are you observing a Linear complexity? Write your observations in the same word file (as that of Program 1) about flat profile that you must have observed and upload it to the lab course page at the end of the lab).

Program 3: There are n number of stairs in a staircase. Assume that you are standing at the bottom of the staircase. You can climb either 1 stair, 2 stairs or 3 stairs at a time. The task is to write a recursive function to count the total number of ways to reach the top of the staircase i.e. the nth stair. For example:
If n=3, there are total of 4 paths to reach top of staircase.

The ways are:
1 step + 1 step + 1 step
1 step + 2 steps
2 steps + 1 step
3 steps

Task: Your task is to complete the missing code in Program3.cpp. Also, measure the time taken for say, 25 stairs and 5 stairs using chrono library. What happens if you give a large input, say 3000?

(**Note:** Write your observations in the same word file (as that of Program 1 and 2) about the resource requirement in case of a recursive procedure and upload it to the lab course page at the end of the lab).

Program 4: You are given an array of non-negative integers and a target sum. The task is to write a recursive function to check if any subset of the given array has the sum equal to the target sum. Below are the two cases, one with success and one with failure.

```
Enter the number of elements: 4
Enter the non-negative elements:
1 3 4 5
Enter the required sum: 8
A subset exists with sum equal to 8
```

```
Enter the number of elements: 4
Enter the non-negative elements:
2 3 5 7
Enter the required sum: 6
No subset exists with sum equal to 6
```

Task:

An incomplete code snippet is given in Program4.cpp. Complete the missing code (?s) and test it with multiple test cases. Note whether the function is tail-recursive or not.

(**Note:** Write your observations about if it is tail-recursive function in the same word file (as that of Program 1, 2 and 3) and upload it to the lab course page at the end of the lab).
