Data Structures and Algorithms

CS210A, ESO207A, ESO211

Semester I, 2012-13, CSE, IIT Kanpur

Programming Assignment 1

Deadline : 23:59 IST on 12th August

1 Efficiency of algorithms does matter even for simple problems

Let F_n denote nth Fibonacci number. Each of you would have written a code for Fibonacci numbers during the course ESC101. Consider a related computational problem whose input is any two integers n and m, and output is ($F_n \mod m$).

In the second lecture of our course, three algorithms for calculating Fibonacci numbers were discussed. The first algorithm was recursive, the second algorithm was iterative and the third algorithm was based on repeated squaring to compute some power of a matrix. You have to design and implement three algorithms $A_{recur}, A_{iter}, A_{matrix}$ for the above problem by suitably employing each of these three algorithms respectively.

The aim of this exercise is to see how efficiently $A_{recur}, A_{iter}, A_{matrix}$ solve the above problem in real time. You have to achieve the following objectives.

- The program has to work for the following ranges of n and m. The integer n can be any positive integer of value up to 10^{18} and m can be any positive integer up to 10^8. As it will turn out, some of these algorithms will start taking too much time as n increases. You have to experimentally determine the largest possible value of input for each of algorithms $A_{recur}, A_{iter}, A_{matrix}$ for which you get the answer with in 1 sec, 10 sec, 1 minute, 10 minutes.
- Plot the time taken by each of the algorithms $A_{recur}, A_{iter}, A_{matrix}$ as a function of n or $\log_2 n$. Analyze the time complexity of each of the above algorithms theoretically (order notation) and then compare with the experimental findings.
- Mention any other useful inference you can draw from this exercise experimentally and/or theoretically.

Motivation for Fibonacci numbers:

Fibonacci numbers have many applications in theoretical as well as applied computer science. They are used in pseudo-random number generators, Fibonacci heap data structure, analyzing efficiency of Euclid's algorithm etc. Fibonacci numbers are also found in natural patterns like flower petals.

2 A clever use of a simple data structure to solve a problem

The aim of this exercise is to make you realize that sometimes a clever use of a simple data structure can be used to design algorithms for problems that otherwise seem unsolvable in limited resources of present day computers. Design and implement an algorithm which receives any positive integer n (with value up to a few hundreds or thousands if possible) and outputs its factorial.

- Plot the time needed by your program to calculate $n!$ versus $\log_2 n$. What inferences do you draw ? Calculate the time complexity of your algorithm theoretically (order notation) and compare it with the experimental findings.
- What are the largest values of n for which your program computes $n!$ in 1 millisecond, 1 sec, one minute, 10 minutes ?
- Mention any other useful inference you can draw from this exercise experimentally or theoretically.

A Motivational story in the background: Suppose you have a friend in the Physics Department of some NIT. He is conducting some research that often requires him to compute the factorials of large numbers (say a few hundreds or thousands). As he wants to save time, he decides to write a program for this task. Although he has had a good exposure to programming in C, he is unable to make such a program. The problem he faces is that one can store and do arithmetic operations only on 64 bit integers. Therefore, the largest factorial that his program is able to compute is 20!.
Since you are doing a course on Data Structures and Algorithms at IIT Kanpur, he seeks your help. As he is conducting an important research, your program should output the exact value of $n!$. No approximations are allowed.
Note: *Hint:* You have studied the data structure, 'Arrays'. Use Arrays wisely in this problem.

Submission guidelines

You will have to submit the code of the above program before 23:59 on 12th August. The demo and viva voce for this assignment will take place in various slots from 5:15 PM to 7PM during the days 13th August to 18th August. The exact details of the submission process and the demo will be mentioned sometime next week.

"The first and most important step towards success is the feeling that we can succeed" –Nelson Boswell