計算Fibnacci Number(費氏數列)的執行時間

Name: 昂偉

No.: f10215009

Date: 10/07/2013

Abstract

Implement fibonacci function in C and print out the execution times of fibonacci function.

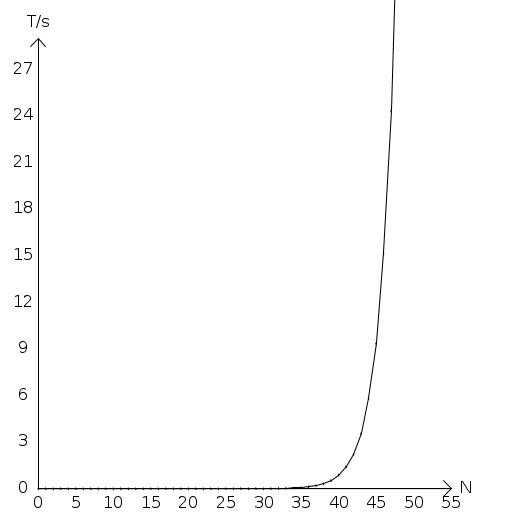
Description

Here, I give a C implementation of fibonacci function. This C program is a recursive version of fibonacci function. Through the analysis of the run time of the program, we could could not only get to know the time complexity of fibonacci algorithm, but also get a rough experience of the process of the analysis of algorithm s. Also we will learn how to get a concrete algorithm's time complexity and running time growth order using generate function method.

How do I get fibonacci function running time? Before calling fibonacci function in function analysis, I record system time using variable start\_time by call clock(), whose job is to return the system's instructions number. After called fibonacci function returns, again, I use variable end\_time to record the system time. So “end\_time – start\_time” is the clocks fibonacci function used. In order to get the running time of fibonacci function, I just need to use CLOCKS\_PER\_SEC to divide “end\_time – start\_time”.

Experiment result

Here are the result of first 52 fibonacci number and their running time in ms: experiment\_data.txt



Complexity analysis

Define as the time needed to run fibonacci function of N, so according to the fibonacci algorithm implemented in C, we could get the recurrence relation:

Using generate function method, we could get the result that , it's consistent with experiment data.

Summary

Through this homework, we learn about the fibonacci algorithm. With the process of analysis of algorithm's running time complexity, we get a method to analysis a general algorithm's time complexity. That is:

(1) Analysis the operations of the algorithm to get the recurrence relation;

(2) Using mathematical method, for example generating function, to get the exact running time expression of the algorithm;

(3) Run one implementation of the algorithm, like C, C++ or C#, to verify the correctness of process of analysis of the algorithm's time complexity.