

106119029 , Algos Lab Model Test

Dipesh Kafle

Question

In mathematical terms, the sequence F_n of Fibonacci numbers is defined by the recurrence relation: $F_n = F_{n-1} + F_{n-2}$. Write an algorithm to compute the n th Fibonacci number in linear time where n is an arbitrary user input. Write the code and run it for random numbers of size $[10, 1000]$ increasing the size linearly. Show in a table the time required to find the number for each of the cases. Plot the data of the table.

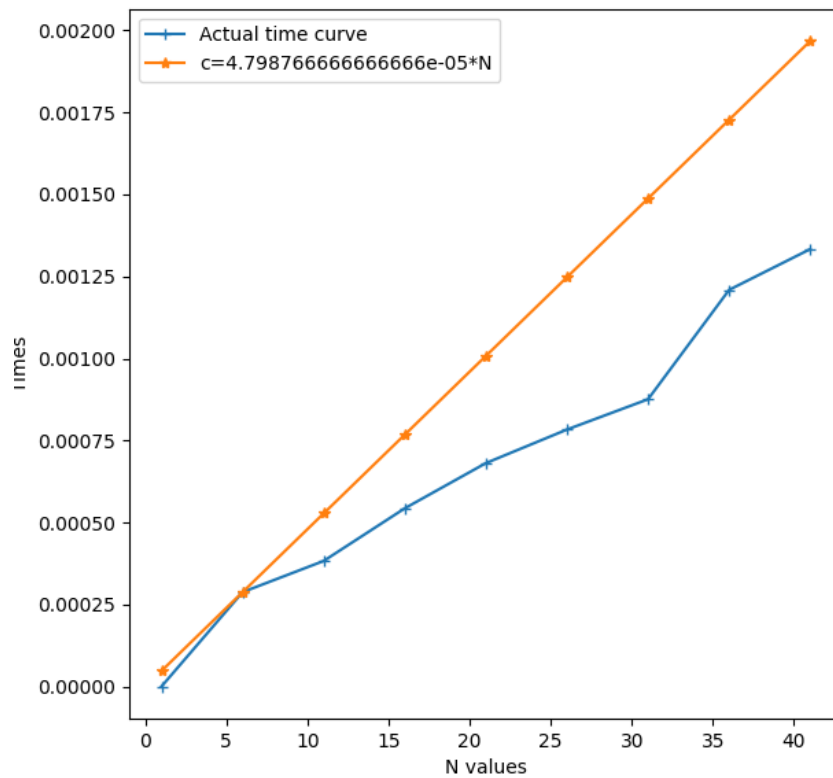
Code

```
1  #include <algorithm>
1  #include <cassert>
2  #include <chrono>
3  #include <iostream>
4  #include <numeric>
5  #include <string>
6  #include <unordered_map>
7  #include <vector>
8
9  using namespace std;
10
11 size_t fibonacci(int n) {
12     // makes sure the number is positive
13     assert(n >= 1);
14
15     if (n <= 2)
16         return n - 1;
17     vector<size_t> fibs(n, 0);
18     fibs[1] = 1;
19     for (int i = 2; i < n; i++)
20         fibs[i] = fibs[i - 1] + fibs[i - 2];
21     return fibs[n - 1];
22 }
23
24 int main() {
25     for (int i = 1; i < 46; i += 5) {
26
27         auto start_time = std::chrono::steady_clock::now();
28         size_t fibonacci_answer = fibonacci(i);
29         auto end_time = std::chrono::steady_clock::now();
30         std::chrono::duration<double> elapsed_time =
31             std::chrono::duration_cast<std::chrono::duration<double>>(end_time -
32                                                                     start_time);
33         cout << i << ':' << fibonacci_answer << ':' << elapsed_time.count() << '\n';
34     }
35 }
```

Output

```
1:0:1.031e-06
6:5:0.000400837
11:55:0.00105771
16:610:0.00136056
21:6765:0.000958866
26:75025:0.000874198
31:832040:0.00111768
36:9227465:0.00144531
41:102334155:0.00259177
```

Graph



- The graph below is upper bound of the fibonacci function.