COL100 Assignment 1

Due date: 9 December, 2020

1 Stair climbing problem

While walking up a staircase with n stairs, you wonder about the number of different ways you can reach the top stair. In one step, you can climb either 1 or 2 stairs (assuming that the 2nd stair exists), and no more than that.

If $n = 1$, there's only one way to reach the top stair, namely by taking a single step.. If $n = 2$, there are two ways: one step of 2 stairs, or two steps of 1 stair each.

1. Let $f(n)$ denote the number of ways we can reach the nth stair. Prove that for $n > 2$, $f(n) = f(n-1) + f(n-2)$.
2. Write a recursive function `climbStair` : `int` $\rightarrow$ `int` that takes input a positive integer n and outputs the number of ways to reach nth stair.
3. For $n > 2$ and $f(n)$ as described above, prove that:

$$f(n) = 2 + \sum_{i=1}^{n-2} f(i)$$

2 Playing with digits

Let n be a positive integer with digits $d_k \dots d_1 d_0$, with d_k being the most significant digit. Define

$$f(n) = 2^k d_k + \cdots + 2^1 d_1 + 2^0 d_0.$$

For example if $n = 132$, then $d_2 = 1, d_1 = 3, d_0 = 2$, so $f(123) = 2^2 \cdot 1 + 2^1 \cdot 3 + 2^0 \cdot 2 = 4 + 6 + 2 = 12$.

1. Find a recursive relation between $f(n)$ and $f(\lfloor n/10 \rfloor)$. Here $\lfloor x \rfloor$ is the greatest integer less than or equal to x.
2. Write a function `modifiedDigitSum` : `int` $\rightarrow$ `int` that takes input a positive integer n and outputs the desired sum $f(n)$.

3 Sum of squares

Given a positive integer n, find the number of positive integers less than or equal to n, that are expressible as sum of squares of two (not necessarily distinct) natural numbers a and b.

For example, if $n = 5$,

- $1 = 0^2 + 1^2$
- $2 = 1^2 + 1^2$
- $4 = 0^2 + 2^2$
- $5 = 1^2 + 2^2$

Thus, there are ~~five~~ four such integers less than or equal to 5.

If $n = 50$, there are 24 such integers.

1. Define an algorithm to find the number of such integers for any given n, and prove its correctness.
2. Implement your algorithm as a function `squaredCount` : `int` $\to$ `int`.

Please note that the above two cases for $n = 5$ and $n = 50$ are just sample test cases for you to verify the correctness of your algorithm. We will check your code on many different cases (all for $n \leq 50$), and not just these two.

4 Summing a series

In the 15th century, Nilakantha published the following infinite series that converges to π:

$$\pi = 3 + \frac{4}{2 \cdot 3 \cdot 4} - \frac{4}{4 \cdot 5 \cdot 6} + \frac{4}{6 \cdot 7 \cdot 8} - \frac{4}{8 \cdot 9 \cdot 10} + \cdots$$

1. Write a function `nilakanthaSum` : `real` $\to$ `real` that takes as input a positive real t and outputs the sum of the series up to the first term whose absolute value is less than t. For example, `nilakanthaSum`$(1.0) = 3 + \frac{4}{2\cdot3\cdot4} = 3.166\ldots$ because $|\frac{4}{2\cdot3\cdot4}| < 1$.
2. Prove the correctness of your algorithm, and of any helper functions.

5 Submission and other logistics

Submission will be on Moodle, similar to Assignment 0. Further details will be added shortly.

Please ask any queries related to the assignment on the COL100 Piazza page (`https://piazza.com/iit_delhi/fall2020/col100`).