Week #12 CFM2103

1. *Euclid's algorithm* is used for finding the *greatest common divisor* (GCD) of two positive integers. The GCD of two such numbers is the largest integer that divides both numbers (without leaving a remainder). The aforementioned algorithm is based on the observation that the GCD of two numbers does not change if the larger number is replaced by its difference with the smaller number. For example,

$$GCD(20, 15) = GCD(5, 15) = GCD(15, 5) = GCD(10, 5) = GCD(5, 5) = 5.$$

   (a) Provide a non-recursive implementation of the above algorithm. You should take care in handling exceptional cases when one of the two numbers is zero, negative, etc.

   (b) Another version of this algorithm is based on using successive integer divisions; if a, $b \in \mathbb{Z}$, then the sequence of computations follows the pattern indicated below:

$$\begin{aligned} a &= q_0 b + r_0 \\ b &= q_1 r_0 + r_1 \\ r_0 &= q_2 r_1 + r_2 \\ r_1 &= q_3 r_2 + r_3 \\ &\vdots \end{aligned}$$

   The remainders keep decreasing with every step but can never be negative, so at some stage during this repeated division process a remainder r_N must be zero (for some $N \geq 0$). This is when the algorithm stops; the GCD of the two numbers is then the *final non-zero remainder* r_{N-1}. Note that we can write

$$GCD(a, b) = GCD(b, r_0) = GCD(r_0, r_1) = \cdots = GCD(r_{N-2}, r_{N-1}) = r_{N-1}\,.$$

   Provide a recursive implementation for this algorithm.

2. The *harmonic numbers* H_n are defined by the formula

$$H_n = 1 + \frac{1}{2} + \cdots + \frac{1}{n}\,, \qquad \text{for } n \geq 1\,.$$

   - Provide a recursive implementation for computing the harmonic numbers.
   - Include a short demo code that uses a for-loop to display on the screen H_n, $\ln(n)$ and $H_n - \ln(n)$ for $n = 10,\ 20,\ \ldots,\ 980$. What do you notice?

3. (a) The bisection algorithm discussed in one of the previous weeks admits a recursive implementation. Provide such a code and test it on the following equations

$$(i)\ \mathrm{e}^x + \ln x = 1000\,, \qquad x > 0\,; \qquad (ii)\ \sin x = \frac{1}{\ln x}\,, \quad 2\pi < x < 5\pi/2\,.$$

April 2022
Dr. Ciprian D. Coman
c.coman@hud.ac.uk

(b) Experiment with your code and the relative accuracy of the approximations obtained.

4. The *binomial coefficients* are positive integers indexed by a pair of integers $0 \leq k \leq n$. They are typically written as

$$\binom{n}{k} \qquad \text{or} \qquad C(n,k)\,,$$

and can be computed by using the formula

$$\binom{n}{k} \equiv \frac{n!}{k!(n-k)!}\,, \tag{1}$$

where the exclamation signs indicate the factorials of the corresponding numbers.

(a) Use (1) to show that

$$C(n,k) = C(n-1,k-1) + C(n-1,k)\,, \qquad \text{for all } n \geq k \geq 1\,. \tag{2}$$

(b) Taking into account that $C(n,0) = C(n,n) = 1$, use the recurrence (2) to compute the binomial coefficients recursively. Make use of a `try...except...` clause in your code.

(c) Improve the above algorithm by using a *memoization* approach and a suitably defined dictionary.

(d) Modify the memoization approach so that instead of a dictionary you make use of a numpy array in which you store the results of the intermediate computations.

5. Consider the integral

$$I_n = \int_0^{\pi/2} \cos^n(x)\,\mathrm{d}x\,, \qquad n = 0,1,2,\ldots\,.$$

- Using integration by parts show that

$$I_n = \frac{n-1}{n} I_{n-2}\,, \qquad n \geq 2\,.$$

- Write a recursive implementation for the calculation of I_n, and then test your code on I_{10} and I_6.

6. Consider the integral

$$I_{n,k} = \int_0^1 x^n \mathrm{e}^{kx}\,\mathrm{d}x\,, \qquad n = 0,1,2,\ldots,k \in \mathbb{N}\,.$$

- Using integration by parts show that

$$I_{n,k} = \frac{\mathrm{e}^k}{k} - \frac{n}{k} I_{n-1,k}\,, \qquad n \geq 1, k \neq 0\,.$$

- Write a recursive implementation for the calculation of $I_{n,k}$, and then test your code on $I_{20,6}$ and $I_{10,5}$.

Dr. Ciprian D. Coman
c.coman@hud.ac.uk