Lab 12 - Recursion

1. Write a recursive function to calculate the sum of the elements of a vector. ★

2. Write a recursive function to compute the n-th element of the Fibonacci sequence. ★

3. Write a recursive function to calculate the values of the Hermite polynomials of degree n defined as follows:

   $H_0(x) = 1$

   $H_1(x) = 2x$

   $H_n(x) = 2xH_{n-1}(x) - 2(n-1)H_{n-2}(x)$, for $n >= 2$

   ★

4. Write a recursive program to generate the subsets of k elements of a set A with a total number of n elements (i.e. combinations of k elements taken from a total of n elements). ★

5. Generate recursively the permutations of the set A having n elements. ★

6. Write a recursive program to generate the partitions of a natural number n. E.g.:

   The partitions of the number 4 are: $\{1,1,1,1\}, \{1,1,2\}, \{1,2,1\}, \{1,3\}, \{2,1,1\}, \{2,2\}, \{3,1\}, \{4\}$. Two partitions may differ either in the value of their elements, or in the order the elements are listed. ★

7. Write a program to solve the problem of the eight queens: i.e. find their placement on the chess board so that they have do not have the possibility to attack each other. ★★

8. There are n cubes, each of them characterized by its color and the length of the edge. Generate, with some of these cubes, a maximum height tower, under the following conditions:

   • a cube is placed on another only if its edge length is less than the edge length of the cube placed bellow it

   • two neighboring cubes must have different colors

   ★★

9. A labyrinth is coded using a $m \times n$ matrix. The passages are represented by elements equal to 1 placed on successive positions on a row or on a column. The requirement is to display all the traces that lead to the exit of the labyrinth, starting from an initial position (i, j). It is forbidden to pass through the same point more than once. ★★★

10. The towers of Hanoi. Consider three vertical pegs A, B, C, and n disks of different diameters. Initially, all the disks are placed on peg A, in descending order of their diameters (with the biggest at the bottom and the smallest at the top). The problem is to move all the disks from peg A to peg C, using peg B as intermediary, under the following conditions:

    • at each step you can move only one disk: the disk located at top of one of the three pegs

    • it is forbidden to put a disk of a greater diameter on a disk having a smaller diameter

    • at the end, the disks must all be on peg C in the same order they were on peg A at the beginning

★★★

References

- Pb. 3-10 [1]

[1]   Iosif Ignat & Marius Joldoș. *CP Laboratory Guide 12: Recursion.*