Mila University Center

Institute of Mathematics and computer science

2nd ^{year} L MD computer science

ASDD1 Module

TD 02: Linked Lists

Exercise 01:

Write the algorithm that allows entering a linked list of N integers and then display: its elements, its length, the sum of its elements, its minimum, and its mirror. At the end, the algorithm must free all the memory space reserved by the list L.

Exercise 02: Write the following recursive operations on linked lists of integers.

1. **Belong (L, x)** allowing to search for an element in the list L.
2. **Max(L)** that returns the largest element of L.
3. **Is_sorted** (L) to check whether a list is sorted in ascending order or not.

Exercise 03: write a function (recursive and iterative) Merge (L1, L2) that takes two lists sorted in ascending order and returns a sorted list, in the same order, containing the elements of both lists.

Exercise 04: Write an iterative and a recursive version for the following functions on linked lists of integers.

1. Insert(pos,x,L): Allows you to insert x into position ***pos***.
2. Delete_pos (L, pos): deletes the element in the position ***pos***.
3. InsertSort (L, x): allows to insert x into the sorted list L while keeping the order of the elements.

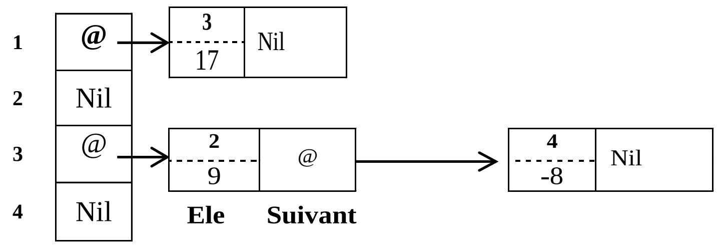
Exercise 05

A sparse matrix is a matrix in which the majority of elements are zero. A sparse matrix can be represented, taking into account only the non-zero elements, by using an array of N lists (N is the number of rows in the matrix). Each row of the matrix is represented by an ordered (according to the rank of the column) linked linear list of the non-zero elements. Each element of the list contains the index (j) of the column and the value (v) of the non-zero element of the matrix.

Example: below is an example of the representation of a matrix **A** by an array of lists ***Tlists*** .

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & 0 & 17 & 0 & 0 \\ 2 & 0 & 0 & \mathbf{0} & 0 & 0 \\ 3 & 0 & 9 & 0 & -8 & 0 \\ 4 & 0 & 0 & 0 & 0 & 0 \end{array}$$

A (4*5)



Tlistes

1. Give the necessary declarations.

2. Write a procedure ***display* (Tlists , N, M)** that takes as input an array *Tlists* representing a sparse matrix of N*M elements and displays all the elements, zero and non-zero, of this matrix.

3. Write a function ***percentage* (Tlists , N, M)** taking as input an array *Tlists* representing a sparse matrix of N*M elements and which returns the percentage of zero elements in the matrix.

4. Write a ***diagonal function* (Tlists , N)** admitting as input an array **Tlists** representing a square matrix of N*N elements and which allows to check if this matrix is diagonal or not. A square matrix is said to be diagonal if all elements off the diagonal are zero.

Exercise 06: additional

Let L be a linked list of students for the ASD3 module. Each student is represented by his number, his name, his first name, TD note, TP note and a control note.

1. Do the necessary modular splitting and write the algorithm to capture a list of N students and then display the list of students who obtained the module.
2. Write the function Postponed (L) to create a list containing the students who did not obtain the module.

Exercise 7: additional

We consider the construction of a list of prime numbers less than or equal to a given integer n. To construct this list, we begin, in a first phase, by adding all the integers from 2 to n, starting with the largest and ending with the smallest, which will be at the head of the list. We then successively consider the elements of the list in ascending order and remove all their strict multiples. Write the algorithm and all necessary functions.