Birla Institute of Technology & Science, Pilani
2nd Semester 2016-17 - CS F211 – Data Structures and Algorithms

---

Lab 4 (Evaluation 1) : 9th Feb 2017

Time: 170 minutes **Marks: 8 + 22 = 30**

Instructions:

- *This test consists of two problems (Problem 1 and Problem 2) specified in two different files.*
- All input expressions should be read from stdin (scanf) and output should be printed on stdout (printf).
- For first 150 minutes, only a subset of test cases will be visible to students after submitting the code on the portal. Only in last 20 minutes, all test cases will be made visible.
- At the end of 170 minute period, the online system will stop evaluating the submissions but it will accept it for additional 10 minutes. At the end of 180 minute period, it will stop accepting the submissions.
- Only the last submission by the student for each problem will be considered for evaluation, irrespective of earlier correct submission.
- Assuming that a problem contains M marks, in case of (Run-error/Compiler-error/Timelimit-error), evaluation will be done for M/2 marks only.
- Total marks of each problem contains some marks for modularity and proper structuring of code.
- All submitted source code will be later checked manually by the instructor and final marks will be awarded. Any case of plagiarism and/or hard coding of test cases will fetch 0 marks for the problem/evaluation component.
- Make sure to return 0 from the main() function in case of normal termination.

Problem 2 of 2

Expected Time: 100 minutes **Marks: 22**

Problem Statement

A sparse matrix is a matrix of mostly null values. For instance, a matrix of M * N numerical values is said to be sparse if the number of non-zero values is close to M + N rather than M * N. In such a case, instead of using M *N locations to store the matrix one may use a design structure that uses space proportional to M + N instead by storing only the non-zero elements.

One such data structure is an array of (linked) lists: the matrix is an array of rows, where each row is a linked list of pairs of the form <column number, non-zero value>.

Implement following functions for "array of linked lists" representation of sparse matrix. Key value of all functions will be same as in problem 1. Input format of all functions except readMatrix and printMatrix will also be same as problem 1.

- **readMatrix** function will be modified as follows:
  - Key: 0
  - Format:

1 matID				
R_i	C_p val_1	C_{p+x} val_2	C_{p+x+y} val_3	-1
R_{i+r}	C_q val_4	C_{q+w} val_5	-1	
-2				

  - Description:
    - "1" shows start of a new matrix.
    - "matID" represents Matrix ID, which will be unique integer among matrices.
    - Each next line, until "-2" is encountered, will represent a row of the matrix.
    - Each row, will start with a row number (in strictly increasing order, starting with 1).
    - On each line, row number will be followed by tab separated pair of values (C and val). C represents the column number (starting with 1) and val represents the non-zero value. C and val will be separated by a space. "-1" will represent end non-zero values in the row.
    - Call printMatrix in the end.
- **printMatrix** function should print zero as well as non-zero values in M rows and N columns, if M is the maximum row number and N is the maximum column number. Each row should be printed on a new line with tab separated values.
- **rcMul:** same format as problem 1.
- **rmMul:** same format as problem 1.
- **mmMul:** same format as problem 1.
- **memProf:** same format as problem 1.

Test Case 1:

Input	Output
1 **1** 1 1 1 2 2 4 3 -1 2 1 2 4 1 -1 3 1 1 -1 -2 *1* **2** 1 1 2 3 3 4 1 -1 2 3 1 4 2 -1 3 1 1 2 1 3 1 -1 4 1 2 3 1 5 5 -1 -2 *2* **1** 2 **2** 3 *3* **1** 3 **2 1** *4* **1 2 101** *5* **101** *1* **20** 1 1 1 2 2 -1 2 1 3 -1 3 2 6 -1 4 -1 5 -1 -2 *4* **101 20 102** *5* **102** *-1*	1 2 0 3 2 0 0 1 1 0 0 0 2 0 3 1 0 0 0 1 2 0 1 1 1 0 0 2 0 1 0 5 7 2 0 3 1 0 8 0 8 5 15 6 0 7 2 5 2 0 3 1 0 1 2 3 0 0 6 0 0 0 0 8 64 6 54 2 22