Data Structure & Algorithms
CS210A
Semester II, 2015-16, CSE, IIT Kanpur

Programming Assignment II

Deadline : 11:59pm of 25th January

Note: This programming assignment consists of two problems and you have to do **only one of the two problems**. First one is a hard problem compared to the second one. So the marks distributions are not same. **The first problem has 100% marks** and **the second has only 65%** marks of the total assisnment.
In case you face any problem in understanding these problems or their solution, you may contact the instructor to get some help or hint.

1 Data structure for compact representation of sparse matrices

The simplest way to represent a $n \times n$ matrix M is by a 2-dimensional array. A matrix M is said to be a sparse matrix if the nonzero entries in M are very few. Storing a sparse matrix using 2-dimensional array is not a compact way to store. However, any alternate data structure should be such that it facilitates efficient execution of various algorithms on matrix. The goal of this problem is to make you realize that there is a very elegant link based data structure for storing matrices which achieves compactness. A skeleton of the data structure you need to develop is shown in Figure 1.

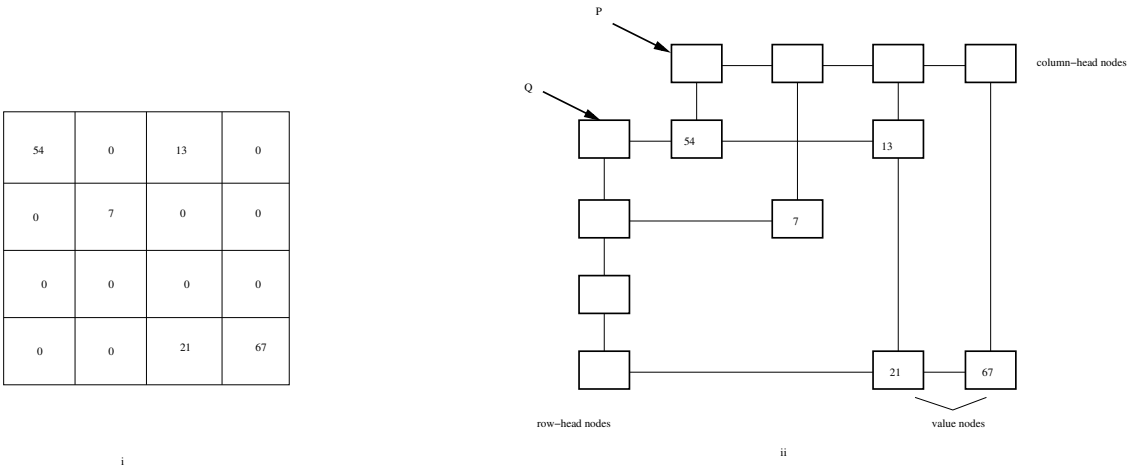

Figure 1: (ii) shows a skeleton the link based data structure for the sparse matrix shown in (i)

There are the following features of this data structure. The data structure for an $n \times n$ matrix will consist of exactly $2n + m$ nodes where m is the number of non zero entries in the matrix. Each node will have identical structure, that is, a row-head node, a value node, and a column-head node will have identical structure (your implementation has to use some way to distinguish these nodes).

The matrix will be accessed by two pointers P and Q pointing to the lists of column-head nodes and row-head nodes respectively (see Figure 1).

As part of the assignment, you have to achieve the following objectives.

- Design the data structure with the above specification.
- You have to develop routines to read entries of matrices and store them in your data structure. You may assume that the non zero entries will be provided to you in the following order: First the nonzero entries of 1st row will appear, and then the nonzero entries of second row will appear, and so on. The entries within a row will appear in the increasing order of their columns (see the sample input below).
- You have to design an algorithm to multiply two $n \times n$ matrices which are stored in the data structure designed above.

Input: The first line will represent n. Thereafter the nonzero entries of matrices will appear in the following format. Each line will consist of four numbers. The first number will represent the matrix (first or second). The second and the third numbers will represent the row and column of the entry. The fourth number will represent the value of the entry. For example, 1 4 2 36 means that there is a nonzero entry in the 4th row and 2nd column of the first matrix and its value is 36. First all the nonzero entries of first matrix appear and then the nonzero entries of the second matrix appear. Finally a line consisting of a single 0 means the end of the input. You may assume that input to your program will indeed be a valid input. Both the indices of row and column number start from 1.

Output : You have to print the output matrix after multiplication. The output should be in the above format given as well. Do not give any matrix number for the resultatant matrix in the output. For example, 4 2 36 means that there is a nonzero entry in the 4th row and 2nd column of the output matrix and its value is 36.

Sample Input:

```
3
1 1 3 25
1 2 1 33
1 2 2 14
1 3 1 8
2 1 1 45
2 1 2 79
2 3 1 109
2 3 3 56
0
```

Sample Output:

```
1 1 2725
1 3 1400
2 1 1485
2 2 2607
3 1 360
3 2 632
```

2 Data structure for Big Polynomial Representation

Linked list also can be used for representing polynomials. This assignment deals with representing two polynomials given in the input using linked list and multiply them and give the output polynomial as linked list as well. The nodes can be used to store the coefficients of the polynomial in an ascending order i.e for a nth order polynomial the first node can represent the coefficient for the 0th order term and the last node can represent the coefficient for the nth order term(similar for descending order). You also can store the order of the particular term in the nodes if you like.

As part of the assignment, you have to achieve the following objectives.

- Using linked list represent the 2 polynomials.
- Design the algorithm to multiply this two polynomials and the result should be stored in another linked list.

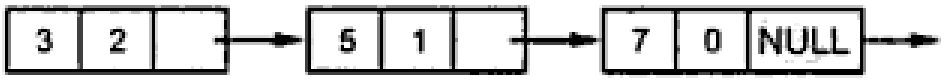

Figure 2: Shows a representation of a 2nd order polynomial starting with the 2nd order term and ending with the 0th order term

.

Input : The first line contains 2 integers representing the order of the two polynomials. The next two lines contain the coefficient of the polynomials from 0 to the given order respectively. For example, if the first polynomial is of order 3,then the 2nd line will contain coefficients like 8 0 1 17 where coefficient of x^0 is 8, coefficient of x^1 is 0, coefficient of x^2 is 1 and coefficient of x^3 is 17.

Output : The output is the resulting polynomial after multiplication represented in the same format as the input.

Sample Input :

```
3 2
0 1 3 14
1 2 0
```

Sample Output :

```
0 1 5 20 28
```