

# COP3530

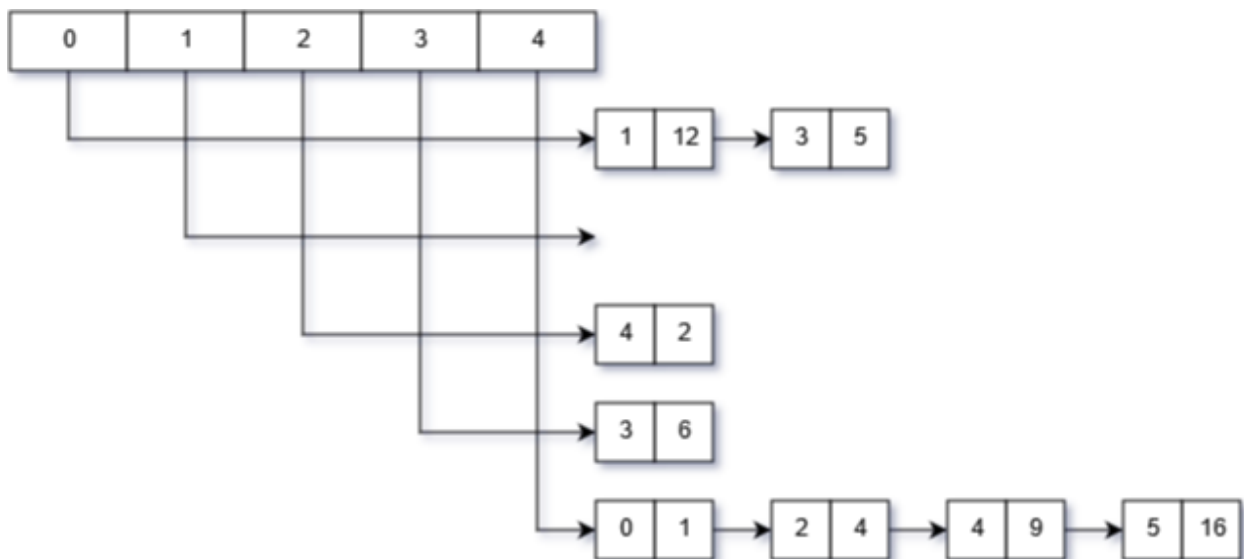
## **Homework #3**

## Sparse Matrices.

A sparse matrix is a matrix for which the majority of elements are zero. Hence, in order to efficiently work with sparse matrices, usually, only non-zero elements are stored instead of storing the whole matrix in a 2D array.

In this assignment you have to implement a data structure for efficiently storing sparse matrices using an array of linked lists. Each row of the matrix will be stored in a linked list, and each non-zero element of that row will be stored in the linked list using its column number and its value. For example, take a look at the following  $5 \times 6$  matrix and how it will be stored in this data structure. Note that the zeros should not be stored in the linked lists.

$$\begin{bmatrix} 0 & 12 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 6 & 0 & 0 \\ 1 & 0 & 4 & 0 & 9 & 16 \end{bmatrix}$$



## Reading the matrices from input.

The format by which the matrices will be given to your program is as follows.

On the first line a number,  $N$ , showing the number of rows in the matrix. Then for each row, first a number,  $K$ , showing the number of elements in that row, and in the next  $K$  lines, two number showing the column number and the value of each element in that row. For example, below you can see how the aforementioned  $5 \times 6$  matrix will be given to your program. Note that all indices are zero-based. (Blue statements are only for your information, they are not part of the input).

5		Number of rows
2		Number of elements in the 0 <sup>th</sup> row
1 12		Column 1, value 12
3 5		Column 3, value 5
0		Number of elements in the 1 <sup>st</sup> row
1		Number of elements in the 2 <sup>nd</sup> row
4 2		Column 4, value 2
1		Number of elements in the 3 <sup>rd</sup> row
3 6		Column 3, value 6
4		Number of elements in the 4 <sup>th</sup> row
0 1		Column 0, value 1
2 4		Column 2, value 4
4 9		Column 4, value 9
5 16		Column 5, value 16

In this format, the elements in each row are guaranteed to be sorted according to the column number.

## Operations.

Your program should be able to do two operations on the matrices.

**Addition:** Your program should be able to add two sparse matrices. Note that adding two matrices means adding corresponding elements together. For example:

$$\begin{bmatrix} -2 & 0 & 3 \\ 0 & 0 & 1 \\ 2 & -4 & 0 \end{bmatrix} + \begin{bmatrix} 4 & 2 & 0 \\ 0 & 1 & -2 \\ 0 & -2 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 2 & 3 \\ 0 & 1 & -1 \\ 2 & -6 & 0 \end{bmatrix}$$

If  $A$  and  $B$  are two matrices with  $N$  rows, with  $k_A$  being the total number of elements in  $A$  and  $k_B$  being the total number of elements in  $B$ , your program is expected to compute  $A + B$  in  $O(N + k_A + k_B)^1$ .

**Search:** Your program should be able to search a matrix for a certain value and find all of the rows and columns for which the value is encountered inside of the matrix. For example, consider the following matrix:

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 3 & 0 & 0 & 2 \\ 1 & 0 & 2 & 0 \\ 0 & 4 & 0 & 0 \end{bmatrix}$$

If we search for the value 3 the program should find out that the matrix contains 3 at (1,0). If we search for the value 1 the program should find out that the matrix contains 1 at (0,0), (0,2) and (2,0).

If  $A$  is a matrix with  $N$  rows and  $k$  elements in total, your program is expected to search for a value in  $O(N + k)^2$ . Note that since the matrix is sparse, your program is not expected to be able to search for 0.

---

<sup>1</sup> The reason  $N$  is being added to the number of elements is that, even if there is only one element in the matrix, you still need to check all of the linked lists to find that element, so any operation you do with the matrix will be at least  $O(N)$ .

<sup>2</sup> The reason  $N$  is being added to the number of elements is that, even if there is only one element in the matrix, you still need to check all of the linked lists to find that element, so any operation you do with the matrix will be at least  $O(N)$ .

Your program should read two matrices and compute their addition. Then read a number of values, search for them in the resulting matrix, and print the results of the search operations.

First, there are two matrices according to the format given in the previous pages. The two matrices are guaranteed to have the same number of rows. Then in the next line a number,  $M$ , which is the number values you are going to search for. In the next  $M$  lines there are the values for which your program should do the search operation.

Your program should print the output of each search operation in a single line, with row and column numbers separated with a space. Note that you should print the output of each operation sorted according to row number (and column number if on the same row).

Number of values to search for

The values to search for

Sample Output (Blue statements are only for your information. They are not part of the output):

1 2		Output for 1		The		3 -2 3
		Output for -1		result of	=>	-2 0 1
0 1 1 0		Output for -2		addition		2 0 4
0 0 0 2		Output for 3				

Further explanation about the output of search operation: As you can see in the above sample, 1 can only be found at position (1,2), i.e. row number 1 and column number 2, so the output of searching for 1 is 1 2. On the other hand, -2 can be found at positions (0,1) and (1,0), so the output for searching for -2 is 0 1 1 0. Note that the positions are sorted according to row number of column number, in other words, you should search the matrix from top row to the bottom row, and in each row, from left column to the right column. As you can see in the above example, if the matrix does not contain the given value you should just print a blank line. Also note that all of the positions are on a single line separated with spaces, for example, If a value has been found at positions (0, 1), (0,3), (2,2) and (4,0) the output for that search operation would be 0 1 0 3 2 2 4 0.

## Notes.

- All numbers are signed integers.
- Your program should read a matrix with  $N$  rows and a total of  $k$  elements in  $O(N + k)$ .
- You should not store zero elements.
- All indices are zero-based.
- Your program is not expected to search for 0 in the matrices.
- **No late submission will be accepted.**
- The output of each search operation should be sorted with respect to row number. If there are multiple occurrences of the value in the same row, they should be sorted according to the column number.
- For any questions, please use the Discussion Section of Canvas.
- Your code is going to be compiled with GCC. It's your responsibility to make sure there are no compile errors. If your code generates compile errors you will receive no credit.
- You should submit a single cpp file. Do not archive your source code.