**CSCI 520.001**

**Programming Assignment #2**

The agreement for two binary vectors A and B, each of size n+1, is defined as the following:

int sum=0; for (int i=0; i<=n; i++) if (A[i]==B[i]==1) sum++; (Calculation 1)

In this assignment, you write C++ programs that compute the agreement for two given vectors (see Calculation 1 above). In these programs, in vectors A and B only nonzero elements (elements that are 1) are stored. We give you a function that generates input for a test case for your programs. Both correctness and efficiency of your programs are important.

Please see Attachment #1 for this assignment. However, **you implement a linked list** version of the program started in Attachment #1, which uses arrays. That is, instead of an array, you use linked lists to store the non-zero elements of the vectors. This involves changing struct definition to include a link, and writing a new agreement() function. Keep the same input vectors, but you need to write a new generate\_row\_and\_col() function to create linked lists.

Turn in your work in the drop box for Assignment 2 (for Week 2) in eCollege course for CSCI 520 (not for the lab).

YOUR WORK IS DUE 11:59PM ON THURSDAY, SEP. 11. YOU ARE ALSO REQUIRED TO SHOW A DEMONSTRATION OF THIS PROGRAM IN THE LAB SESSION ON SEP. 12 IF YOU ARE ENROLLED IN THE LAB.

Good Luck

Abdullah N. Arslan

/\* ATTACHMENT 1: \*/

#include <stdio.h>

#include <iostream.h>

/\*

For Assignment #2

You are asked to write a function that calculates the agreement for two vectors, a row and a column

Normally this agreement can be calculated as follows:

int sum=0; for (i=0;i<=n; i++) if (A[i]==B[i]==1) sum++;

(Calculation 1)

But, you will need to do it differently since zeros are not stored in vectors

function generate\_row\_and\_col() below is provided only for testing. Your function should not be designed for a fixed pair of row or column vectors;

row is a row vector of size at most n,

col is a (column) vector of size at most n,

each of these vectors stores only 1’s

here we use an array to store the indices at which the element is 1

e.g. for row vector [0,0,1,0,0,1,0], which has 1’s at indices 2, and 5, respectively

row[0]=2;

row[1]=5;

row[2]=-1 (-1 in x marks the end of non-zero elements in row)

Please note that all other elements (not stored in row) are zeros

Similarly let col be the following vector [0,0,0,0,0,1,0], then

col[0]=5;

col[1]=-1;

For this pair of row and col, the agreement is 1, because row[1]=col[0]=5, and all other terms in Calculation 1 (see the top of the page) are zeros

\*/

static const int n=1000; /\* vector size limit \*/

struct element {

int x; /\* original index of non-zero array element \*/

int val ; /\* integer non-zero value at index x \*/

} ;

int row[n] ;

int col[n] ;

int i;

void generate\_row\_and\_col() {

for (i=1; i<n/4; i++) row[i]=4\*i;

row[0]=1;

row[n/4]=-1;

for (i=1; i<n/5; i++) col[i]=5\*i;

col[0]=1;

col[n/5]=-1;

}

int agreement()

{

/\* calculate the agreement for row and col;

output the result

\*/

}

int main()

{

generate\_row\_and\_col() ;

agreement();

return 0;}