Lab Assignment 1: Advanced Algorithms

Jainil Shah, Dipin Garg, Harsh Kushwaha,
Harsh Kumar Jain, Rahul Kumar

23rd January 2023

1 Pseudocode

```
function luDecompostionRecurssive(Argument marix A):
    Let N = A.rows
    Let A11  = left-topmost element of A

    Base condtion:
    if dimension of A is 1X1:
        return [[1]] and matrix A

    define new matrix:
    aDash as submatrix of A dimension((N-1)X(N-1)) by
       removing top row and top column
    wT row-matrix dimension((1)X(N-1)) contains element
       from first row excluding the first element
    v column-matrix dimension((N-1)X(1)) contains element
        from first column excluding the top element

    calculate:
    NewSubmatrix = aDash-(martix multiplication of(v,wT))
       /A11

    Recurssive Function Call:
    Let submatrix(dimension((N-1)X(N-1))) LDash,Udash =
       output from the function luDecompostionRecurssive(
       argument NewSubmatrix)

    construct a matrix Lower triangular matrix L by
    1) horizontally joining matrix V/A11 on left of Ldash
    2) Vertically joining a matrix with dimension(1X(N))
       with all element zero on top of Matrix obtained in
        step 1
    3) make top leftmost element 1
```

```
    construct a matrix Upper triangular matrix U by
    1) horizontally joining matrix with dimension((N-1)X1
        ) with all element zero on left of Udash
    2) horizontally adding element A11 on left of WT
        matrix
    3) vertically joining the matrix obtained in step two
         on top of matrix obtained on step 1

    return L and U


Define a function Det(argument matrix A):
    Let Lower triangular matrix be L and Upper triangular
         matrix be U
    L,U  = output from the function
        luDecompostionRecurssive(argument A)
    return product of digonal elements of L and U



Read the text from input file
Let N and M defines number of vertex and edges
Let Q be incidence matrix with dimension (NXM) initialize
     all values to zero
assign value:
for i=1 to M edges:
    Let u and v are vertices connection the edge i
    assign Q[u-1][i] = 1
    assign Q[v-1][i] = -1

Let L be Laplacian matrix with dimension (NXN)
define as matrix multiplication of Q and Transpose(Q)

let i = generate any random number from 0 to N-1
Let j = i
delete i-th row  and j-th column from Laplacian matrix L
return  the value of product of det(argument L) and (-1)
    ^(i+j)
```

2 Functionality

Firstly, we are reading the input from the input file. In input, we are given ‘N’: number of vertices, ‘M’: number of edges and after that we are given M values of ‘u’ and ‘v’ such that there is an edge from ‘u’ to ‘v’.

We are creating the incidence matrix (Q) using the input and then we are finding Laplacian matrix (L) using the matrix multiplication of Q and Q-transpose.

We are now left with finding the cofactor of any element of the matrix L as it will give the number of the spanning trees of the given graph. We generate two random integer values (between 1 and N-1) and now we will find the cofactor of the element Lij. For this, we delete the i-th row and j-th column of the matrix L and then use LU decomposition to find the determinant of the matrix.

To find determinant using LU decomposition, we create a function that uses recursion to calculate the LU decomposition of a matrix A.
It takes one matrix as its input and return one lower triangular and one upper triangular matrix as its output.

For a matrix A, consider
N: number of rows
A11: Top-leftmost element of A
aDash: submatrix of A after deleting first row and first column
wT: first row of matrix A excluding A11
v: first column of matrix A excluding A11

We define a new submatrix as aDash – (matrix multiplication of v and wT)/A11. We find the LU decomposition of this submatrix by recursion and call it lDash and uDash.

We now construct our lower triangular matrix by

1. Horizontally stacking matrix v/A11 to the left of lDash.

2. Vertically stacking a matrix of zeros of dimension 1xN to it.

3. Top-leftmost element set to 1.

Similarly, we construct our upper triangular matrix by

1. Horizontally stacking matrix of zeros of dimension (N-1)x1 to the left of uDash

2. Horizontally stacking element A11 to the left of wT matrix.

3. Vertically stacking the matrix obtained in step 2 to the top of that obtained in step 1.

We return these two matrices. For the base case; when order of matrix is 1x1, we return lower triangular matrix as a 1x1 matrix with value of element 1 and upper triangular matrix as A itself. The determinant of upper/lower triangular matrix is equal to the product of its diagonal entries.

So, we calculate the products of diagonal entries of the two triangular matrices and then multiply them to get the determinant of matrix A. In this way, we can calculate determinant of a matrix in polynomial time. Lastly, we multiply the value of the determinant with (-1)$\hat{(}$i+j) to get the number of spanning trees.

3 Complexity Analysis

In the recursive function for computing LU Decomposition, the number of steps apart from the recursive call are in the order of O(n$\hat{2}$), due to the calculation of dot product in the following step:

```
LU = aDash - np.dot(v, wT)/a11
```

The recursive call calculates LU decomposition for matrix in the order of n-1, and hence the recurrence relation we obtain is:

$$T(n) = T(n-1) + n^2$$

To solve the recurrence relation, replacing T(n-1) by T(n-2)+(n-1)$\hat{2}$ gives

$$T(n) = [T(n-2) + (n-1)^2] + n2 = T(n-2) + (n-1)^2 + n^2$$

and similarly

$$T(n) = T(n-2) + (n-1)^2 + n2 = T(n-3) + (n-2)^2 + (n-1)^2 + n2$$
$$= T(n-4) + (n-3)^2 + (n-2)^2 + (n-1)^2 + n^2$$

$$T(n) = T(n-k) + (n-k+1)^2 + (n-k+2)^2 + \ldots + (n-1)^2 + n^2$$

Letting k=n,

$$T(n) = T(0) + 1^2 + 2^2 + 3^2 + \ldots + n^2$$

Since

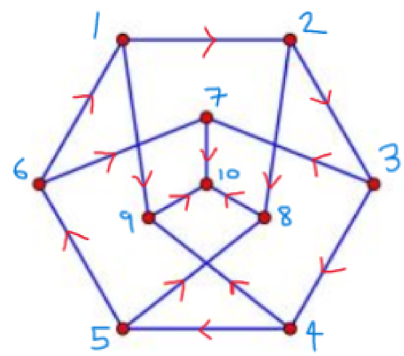
$$1^2 + 2^2 + 3^2 + \ldots + n^2 \leq n^2 + n^2 + n^2 + \ldots + n^2 = n^3$$

$$T(n) = O(n^3)$$

Now, the time complexity of the program will be equal to the time complexity of the LU decomposition function, since the rest of the steps except LU decomposition are of the order of O(n$\hat{2}$) or lower.

4 Laplacian and LU Matrices

The graph was labelled the following way:



The laplcian obtained for the matrix is as follows:

```
[[ 3. -1.  0.  0. -1.  0.  0. -1.  0.]
 [-1.  3. -1.  0.  0.  0. -1.  0.  0.]
 [ 0. -1.  3.  0.  0. -1.  0.  0.  0.]
 [ 0.  0. -1. -1.  0.  0.  0. -1.  0.]
 [ 0.  0.  0.  3. -1.  0. -1.  0.  0.]
 [-1.  0.  0. -1.  3. -1.  0.  0.  0.]
 [ 0.  0. -1.  0. -1.  3.  0.  0. -1.]
 [ 0. -1.  0. -1.  0.  0.  3.  0. -1.]
 [-1.  0.  0.  0.  0.  0.  0.  3. -1.]]
```

The decomposed L matrix is as follows:

```
[[ 1.    0.    0.    0.    0.    0.    0.    0.    0.  ]
 [-0.33  1.    0.    0.    0.    0.    0.    0.    0.  ]
 [ 0.    0.    1.    0.    0.    0.    0.    0.    0.  ]
 [ 0.    0.   -0.    1.    0.    0.    0.    0.    0.  ]
 [-0.33 -0.12  0.12  0.38  1.    0.    0.    0.    0.  ]
 [ 0.    0.    1.    3.    4.    1.    0.    0.    0.  ]
 [ 0.   -0.38  0.38  1.12  2.    0.5   1.    0.    0.  ]
 [-0.33 -0.12  0.12  1.38  2.    0.5   1.    1.    0.  ]
 [ 0.    0.   -0.   -0.   -0.   -0.    0.67 -0.33  1.  ]]
```

The decomposed U matrix is as follows:

```
[[ 3.    -1.     0.     0.     0.    -1.     0.    -1.     0.  ]
 [ 0.     2.67  -1.     0.     0.    -0.33   0.    -0.33   0.  ]
 [ 0.     0.    -1.     3.    -1.     0.     0.    -1.     0.  ]
 [ 0.     0.     0.    -1.     3.    -1.     0.     0.     0.  ]
 [ 0.     0.     0.     0.    -2.     3.    -1.    -0.25   0.  ]
 [ 0.     0.     0.     0.     0.   -10.     7.     2.    -1.  ]
 [ 0.     0.     0.     0.     0.     0.    -1.5   -0.25  -0.5 ]
 [ 0.     0.     0.     0.     0.     0.     0.     2.5    0.  ]
 [ 0.     0.     0.     0.     0.     0.     0.     0.     3.33]]
```

The number of spanning tress came out as the following:

```
2000.0
```