TÜRKİYE CUMHURİYETİ

YILDIZ TEKNİK ÜNİVERSİTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

ALGORİTMA ANALİZİ DÖRDÜNCÜ ÖDEV RAPORU

Öğrenci No: 20011023

Öğrenci Adı Soyadı: Mehmet Alperen Ölçer

Öğrenci E-Posta: alperen.olcer@std.yildiz.edu.tr

VİDEO LİNK : https://youtu.be/u-KztMEhUQc

Ders/Grup: BLM3021 Algoritma Analizi/ 1.Grup

Ders Yürütücüsü

PROF DR MİNE ELİF KARSLIGİL

19 Kasım 2022

YÖNTEM:

Problem : Dosyadan graf okuma.

Çözüm : Struct dizisi ile dosyadan okunan değerlerle graf oluşturma. Structlar nodeları temsil ediyorlar. Struct içinde isim, node numarası, in degrees değeri, M-X-Y değerleri ile filtre boolean'ları, bağlı olduğunu nodeların indexlerin tutultuğu dizi ve bu dizinin boyutu var.

Problem : M değerine göre grafı güncelleme.

Çözüm : In degree değeri M'den küçük olan node'lar için: node'u silindi olarak işaretleme, node'un bağlı olduğu node'ların in degree değerlerini bir azaltma, başka node'lardan bu node'a bağlantı varsa onları koparma işlemleri yapılıyor. Değişiklik yapılamayacak hale gelene kadar devam ediyor.

Problem : X değerinin altında in degree'si olan node'ları işaretleme.

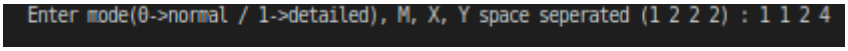
Çözüm : M için grafta eleme yaparken in degree değerlerini güncellediğimiz için bu çözümde sadece node'ların in degree değişkenleri X'ten küçük ise işaretliyoruz.

Problem : Bir node'a ulaşabilen node'ların sayısı Y değerinin altında ise işaretleme.

Çözüm : Güncel graftaki node'lar için Dfs uygulayıp gidilebilen node'ları histogram dizisinde birer arttırıyoruz. Bu işlem sonucunda histogram dizisinde Y'den büyük eşit olan sayısal değerlerin indexleri bize istediğimiz özellikteki node'ları veriyor.

UYGULAMA:

DETAILED 1



```
********************

INITIAL GRAPH


Indegrees : 2
Node Num : 1
Name : Michael Jordan
Number of connections : 3
Connected to : 2 3 5

Indegrees : 2
Node Num : 2
Name : Stephen Boyd
Number of connections : 2
Connected to : 1 3

Indegrees : 2
Node Num : 3
Name : Kalyanmoy Deb
Number of connections : 2
Connected to : 1 2

Indegrees : 1
Node Num : 4
Name : David Johnson
Number of connections : 1
Connected to : 6

Indegrees : 1
Node Num : 5
Name : Scott Kirkpatrick
Number of connections : 2
Connected to : 6 7

Indegrees : 2
Node Num : 6
Name : Lieven Vandenberghe
Number of connections : 1
Connected to : 8

Indegrees : 1
Node Num : 7
Name : Fabian Pedregosa
Number of connections : 1
Connected to : 8

Indegrees : 4
Node Num : 8
Name : Jorge Nocedal
Number of connections : 3
Connected to : 9 10 11

Indegrees : 1
Node Num : 9
Name : Clifford Stein
Number of connections : 1
Connected to : 4

Indegrees : 2
Node Num : 10
Name : Stephen Wright
Number of connections : 2
Connected to : 8 12

Indegrees : 2
Node Num : 11
Name : Philippe Salembier
Number of connections : 1
Connected to : 12

Indegrees : 2
Node Num : 12
Name : Robert Stevenson
Number of connections : 3
Connected to : 8 10 11
```

```
********************

GRAPH AFTER M:1 FILTER


Indegrees : 2
Node Num : 1
Name : Michael Jordan
Number of connections : 3
Connected to : 2 3 5

Indegrees : 2
Node Num : 2
Name : Stephen Boyd
Number of connections : 2
Connected to : 1 3

Indegrees : 2
Node Num : 3
Name : Kalyanmoy Deb
Number of connections : 2
Connected to : 1 2

Indegrees : 1
Node Num : 4
Name : David Johnson
Number of connections : 1
Connected to : 6

Indegrees : 1
Node Num : 5
Name : Scott Kirkpatrick
Number of connections : 2
Connected to : 6 7

Indegrees : 2
Node Num : 6
Name : Lieven Vandenberghe
Number of connections : 1
Connected to : 8

Indegrees : 1
Node Num : 7
Name : Fabian Pedregosa
Number of connections : 1
Connected to : 8

Indegrees : 4
Node Num : 8
Name : Jorge Nocedal
Number of connections : 3
Connected to : 9 10 11

Indegrees : 1
Node Num : 9
Name : Clifford Stein
Number of connections : 1
Connected to : 4

Indegrees : 2
Node Num : 10
Name : Stephen Wright
Number of connections : 2
Connected to : 8 12

Indegrees : 2
Node Num : 11
Name : Philippe Salembier
Number of connections : 1
Connected to : 12

Indegrees : 2
Node Num : 12
Name : Robert Stevenson
Number of connections : 3
Connected to : 8 10 11
```

```
********************

NODES AFTER X:2 FILTER ON M:1 FILTERED GRAPH


Indegrees : 2
Node Num : 1
Name : Michael Jordan
Number of connections : 3
Connected to : 2 3 5

Indegrees : 2
Node Num : 2
Name : Stephen Boyd
Number of connections : 2
Connected to : 1 3

Indegrees : 2
Node Num : 3
Name : Kalyanmoy Deb
Number of connections : 2
Connected to : 1 2

Indegrees : 2
Node Num : 6
Name : Lieven Vandenberghe
Number of connections : 1
Connected to : 8

Indegrees : 4
Node Num : 8
Name : Jorge Nocedal
Number of connections : 3
Connected to : 9 10 11

Indegrees : 2
Node Num : 10
Name : Stephen Wright
Number of connections : 2
Connected to : 8 12

Indegrees : 2
Node Num : 11
Name : Philippe Salembier
Number of connections : 1
Connected to : 12

Indegrees : 2
Node Num : 12
Name : Robert Stevenson
Number of connections : 3
Connected to : 8 10 11
```

```
**********

NODES AFTER Y:4 FILTER ON M:1 FILTERED GRAPH


Indegrees : 1
Node Num : 4
Name : David Johnson
Number of connections : 1
Connected to : 6

Indegrees : 1
Node Num : 5
Name : Scott Kirkpatrick
Number of connections : 2
Connected to : 6 7

Indegrees : 2
Node Num : 6
Name : Lieven Vandenberghe
Number of connections : 1
Connected to : 8

Indegrees : 1
Node Num : 7
Name : Fabian Pedregosa
Number of connections : 1
Connected to : 8

Indegrees : 4
Node Num : 8
Name : Jorge Nocedal
Number of connections : 3
Connected to : 9 10 11

Indegrees : 1
Node Num : 9
Name : Clifford Stein
Number of connections : 1
Connected to : 4

Indegrees : 2
Node Num : 10
Name : Stephen Wright
Number of connections : 2
Connected to : 8 12

Indegrees : 2
Node Num : 11
Name : Philippe Salembier
Number of connections : 1
Connected to : 12

Indegrees : 2
Node Num : 12
Name : Robert Stevenson
Number of connections : 3
Connected to : 8 10 11
```

```
**********

INFLUCERS ACCORDING TO M:1  X:2  Y:4


Indegrees : 2
Node Num : 6
Name : Lieven Vandenberghe
Number of connections : 1
Connected to : 8

Indegrees : 4
Node Num : 8
Name : Jorge Nocedal
Number of connections : 3
Connected to : 9 10 11

Indegrees : 2
Node Num : 10
Name : Stephen Wright
Number of connections : 2
Connected to : 8 12

Indegrees : 2
Node Num : 11
Name : Philippe Salembier
Number of connections : 1
Connected to : 12

Indegrees : 2
Node Num : 12
Name : Robert Stevenson
Number of connections : 3
Connected to : 8 10 11
```

DETAILED 2

```
Enter Mode(0->Normal / 1->Detailed), M, X, Y space seperated (1 2 2 2) : 1 2 3 2
```

```
********************

GRAPH AFTER M:2 FILTER


Indegrees : 2
Node Num : 1
Name : Michael Jordan
Number of connections : 2
Connected to : 2 3

Indegrees : 2
Node Num : 2
Name : Stephen Boyd
Number of connections : 2
Connected to : 1 3

Indegrees : 2
Node Num : 3
Name : Kalyanmoy Deb
Number of connections : 2
Connected to : 1 2

Indegrees : 2
Node Num : 8
Name : Jorge Nocedal
Number of connections : 2
Connected to : 10 11

Indegrees : 2
Node Num : 10
Name : Stephen Wright
Number of connections : 2
Connected to : 8 12

Indegrees : 2
Node Num : 11
Name : Philippe Salembier
Number of connections : 1
Connected to : 12

Indegrees : 2
Node Num : 12
Name : Robert Stevenson
Number of connections : 3
Connected to : 8 10 11



********************

NODES AFTER X:3 FILTER ON M:2 FILTERED GRAPH
```

```
**********************

NODES AFTER Y:2 FILTER ON M:2 FILTERED GRAPH


Indegrees : 2
Node Num : 1
Name : Michael Jordan
Number of connections : 2
Connected to : 2 3

Indegrees : 2
Node Num : 2
Name : Stephen Boyd
Number of connections : 2
Connected to : 1 3

Indegrees : 2
Node Num : 3
Name : Kalyanmoy Deb
Number of connections : 2
Connected to : 1 2

Indegrees : 2
Node Num : 8
Name : Jorge Nocedal
Number of connections : 2
Connected to : 10 11

Indegrees : 2
Node Num : 10
Name : Stephen Wright
Number of connections : 2
Connected to : 8 12

Indegrees : 2
Node Num : 11
Name : Philippe Salembier
Number of connections : 1
Connected to : 12

Indegrees : 2
Node Num : 12
Name : Robert Stevenson
Number of connections : 3
Connected to : 8 10 11



**********************

INFLUCERS ACCORDING TO M:2  X:3  Y:2
```

NORMAL 1

```
Enter mode(0->normal / 1->detailed), M, X, Y space seperated (1 2 2 2) : 0 3 1 5


**********

INFLUCERS ACCORDING TO M:3  X:1  Y:5
```

NORMAL 2

```
Enter mode(0->normal / 1->detailed), M, X, Y space seperated (1 2 2 2) : 0 2 0 3


**********

INFLUCERS ACCORDING TO M:2  X:0  Y:3


Node Num : 1
Name : Michael Jordan
Number of connections : 2
Connected to : 2 3

Node Num : 2
Name : Stephen Boyd
Number of connections : 2
Connected to : 1 3

Node Num : 3
Name : Kalyanmoy Deb
Number of connections : 2
Connected to : 1 2

Node Num : 8
Name : Jorge Nocedal
Number of connections : 2
Connected to : 10 11

Node Num : 10
Name : Stephen Wright
Number of connections : 2
Connected to : 8 12

Node Num : 11
Name : Philippe Salembier
Number of connections : 1
Connected to : 12

Node Num : 12
Name : Robert Stevenson
Number of connections : 3
Connected to : 8 10 11
```

SONUÇ:

->N node'lu bir grafta her node'un da M bağlantısı var ise karmaşıklık O(N*M) olur.

```
void set_indegrees(Node* nodes, int node_count)
{
   int i, j, k;
   for ( i = 0; i < node_count; i++)
      nodes[i].indegrees = 0;

   for ( i = 0; i < node_count; i++)
   {
      if (!nodes[i].M_filtered)
      {
         // Count the number of incoming connections for the current node
         for ( j = 0; j < node_count; j++)
            for ( k = 0; k < nodes[j].num_connected; k++)
               if (nodes[j].connected[k] == nodes[i].num)
                  nodes[i].indegrees ++;
      }
   }
}
```

->N node'lu bir grafta her node'un da M bağlantısı var ise karmaşıklık O(N*N*M) olur.

```
void M_function(Node *nodes, int node_count, int M)
{
// Function to filter nodes based on their in-degree
   // Flag to track if any nodes have been removed
   int removed = 1, i, j, k;

   // Loop until there are no more nodes with an in-degree less than M
   // If no nodes were removed, the loop can be stopped
```

```
    while (removed) {
        i=0;
        removed = 0;
        // Iterate through the array of nodes
        for ( i = 0; i < node_count; i++)
        {
            // Check if the in-degree of the current node is less than M
            if (nodes[i].indegrees < M && !nodes[i].M_filtered)
            {
                // increase indegrees from deleted node's connected array nodes
                for ( j = 0; j < nodes[i].num_connected; j++)
                    nodes[nodes[i].connected[j]-1].indegrees --;

                // remove node from connected arrays
                for ( k = 0; k < node_count; k++)
                    nodes[k].num_connected -= remove_if_exits(nodes[k], i+1);

                nodes[i].M_filtered = true;
                removed = 1;
            }
        }
    }
}
```

->N node’lu bir grafta karmaşıklık O(N) olur.

```
void X_function(Node *nodes, int node_count, int X)
{
    int i;
    for ( i = 0; i < node_count; i++)
        if (nodes[i].indegrees < X)
            nodes[i].X_filtered = true;
}
```

->N node’lu bir grafta karmaşıklık O(N*N) olur.

```
void Y_function(Node* nodes, int node_count, int Y)
{
   int i, j, *histogram = (int*) calloc(node_count,sizeof(int));
   for ( j = 0; j < node_count; j++)
   {
      if (!nodes[j].M_filtered)
      {
         Node node=nodes[j];
         bool* visited = (bool*) malloc(node_count*sizeof(bool));
         for ( i = 0; i < node_count; i++)  visited[i]=false;
         struct stack* s = create_stack(node_count);
         // Push the current source node
         push(s, node);
         while (!is_empty(s))
         {
            // Pop a vertex from stack and print it
            node = peek(s);
            pop(s);
            // Stack may contain same vertex twice. So
            // we need to print the popped item only
            // if it is not visited.
            if(visited[node.num-1] == false)
            {
               // printf("%d -> ", node.num);
               histogram[node.num-1] ++;
               visited[node.num-1] = true;
            }
            // Get all adjacent vertices of the popped vertex s
            // If a adjacent has not been visited, then push it
            // to the stack.
```

```
            for ( i = 0; i < node.num_connected; i++)
                if (!visited[node.connected[i]-1])
                    push(s, nodes[node.connected[i]-1]);
        }
    }
}

for ( i = 0; i < node_count; i++)
    if (histogram[i] < Y && !nodes[i].M_filtered)
        nodes[i].Y_filtered = true;
}
```

VİDEO LİNK : https://youtu.be/u-KztMEhUQc