YILDIZ TEKNİK ÜNİVERSİTESİ

ELEKTRİK – ELEKTRONİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BLM2512 – Veri Yapıları ve Algoritmalar

Ödev - 3

Öğrenci No: 21011055

Ad-Soyad: Emirhan ÖZSARAY

Telefon No: +905418493876

E-Mail: emirhan.ozsaray@std.yildiz.edu.tr

Öğretim Görevlisi: Göksel BİRİCİK

Örnek 1: N = 5

```
Adjacency Matrix:
0 1 1 0 0
0 0 0 1 0
0 0 0 1 1
0 0 0 0 1
0 0 0 0 0

Adjacency List:
Class 1: 3 2
Class 2: 4
Class 3: 5 4
Class 4: 5
Class 5:

Donem 1: Course-1
Donem 2: Course-2 Course-3
Donem 3: Course-4
Donem 4: Course-5

Ogrenci bolumu 4 donemde bitirir.
```

Örnek 2: N = 5

```
Adjacency Matrix:
0 0 1 0 1
0 0 1 1 1
0 0 0 0 1
0 0 0 0 1
0 0 0 0 0

Adjacency List:
Class 1: 5 3
Class 2: 5 4 3
Class 3: 5
Class 4: 5
Class 5:

Donem 1: Course-1 Course-2
Donem 2: Course-3 Course-4
Donem 3: Course-5

Ogrenci bolumu 3 donemde bitirir.
```

Örnek 3: N = 6

```
Adjacency Matrix:
0 0 1 1 0 0
0 0 0 0 0 1
0 0 0 1 0 1
0 0 0 0 0 1
0 1 0 0 0 0
0 0 0 0 0 0

Adjacency List:
Class 1: 4 3
Class 2: 6
Class 3: 6 4
Class 4: 6
Class 5: 2
Class 6:

Donem 1: Course-1 Course-5
Donem 2: Course-2 Course-3
Donem 3: Course-4
Donem 4: Course-6

Ogrenci bolumu 4 donemde bitirir.
```

Karmaşıklık Hesabı: O(N^3)

```
void print(int** adjMatrix, int N){
    int i, j;
    printf("\n\nAdjacency Matrix: \n");
    for(i = 0; i < N; i++){                                 ;N+1
        for(j = 0; j < N; j++){                             ;N*(N+1)
            printf("%d ", adjMatrix[i][j]);
        }
        printf("\n");
    }
}                                                           ;N^2+2N+1
Class* createNode(int j){
    Class* newNode = (Class*)malloc(sizeof(Class));         ;1
    newNode->classNo = j+1;                                 ;1
    newNode->next = NULL;                                   ;1
    return newNode;
}                                                           ;3
int main(){
    int N, i, j, donem = 1, isDone = 0, exit;               ;2
    printf("N: ");
    scanf("%d", &N);
    adjList* list = (adjList*)malloc(sizeof(adjList)*N);;1
    Class* tmp;
    int** adjMatrix = (int**)malloc(sizeof(int*)*N);        ;1
    for(i = 0; i < N; i++)                                  ;N+1
    {
        adjMatrix[i] = (int*)malloc(sizeof(int));           ;N
    }                                                       ;N*(N+1)
    //matris input
    for(i = 0; i < N; i++){                                 ;N+1
        for(j = 0; j < N; j++){                             ;N*(N+1)
            printf("[%d][%d] = ", i+1, j+1);
            scanf("%d", &adjMatrix[i][j]);
        }
    }
    system("cls");
    print(adjMatrix, N);
    //liste oluÅŸturma
    for(i = 0; i < N; i++){                                 ;N+1
        list[i].head = NULL;                                ;N
        list[i].inDegree = 0;                               ;N
        for(j = 0; j < N; j++){                             ;N*(N+1)
            if(adjMatrix[i][j] == 1){                       ;N*(N+1)
                tmp = createNode(j);                        ;N
                tmp->next = list[i].head;                   ;N
                list[i].head = tmp;                         ;N
            }
        }
    }
```

```
    //indegree yerleÅŸtirme;
    for(i = 0; i < N; i++){                             ;N
        for(j = 0; j < N; j++){                         ;N*(N+1)
            if(adjMatrix[j][i] == 1){                   ;N
                list[i].inDegree += 1;                  ;N
            }
        }
    }
    //liste print
    printf("\nAdjacency List: ");
    for(i = 0; i < N; i++){                             ;N+1
        tmp = list[i].head;                             ;N
        printf("\nClass %d: ", i+1);
        while(tmp != NULL){                             ;N^2
            printf("%d ", tmp->classNo);
            tmp = tmp->next;                            ;N
        }
    }
    //dÃ¶nem paylaÅŸtÄ±rÄ±lmasÄ±
    printf("\n");
    while(isDone == 0){                                 ;N
        printf("\nDonem %d: ", donem);
        for(i = 0; i < N; i++){                         ;N*(N+1)
            if(list[i].inDegree == 0){                  ;N^2
                printf("Course-%d ", i+1);
                list[i].inDegree = -1;                  ;N^2
            }
        }
        for(i = 0; i < N; i++){                         ;N*(N+1)
            if(list[i].inDegree == -1){                 ;N^2
                list[i].inDegree--;                     ;N^2
                tmp = list[i].head;                     ;N^2
                while(tmp != NULL){                     ;N^2*(N-1)
                    list[tmp->classNo-1].inDegree--;    ;N^2*(N-2)
                    tmp = tmp->next;                    ;N^2*(N-2)
                }
            }

        }
        exit = 0;                                       ;N-1
        j = 0;                                          ;N-1
        while(j < N && exit == 0){                      ;N^2
            if(list[j].inDegree > -1)                   ;N*(N-1)
                exit = 1;                               ;N
            j++;                                        ;N^2
        }
        if(exit == 0)                                   ;N-1
            isDone = 1;                                 ;1
        donem++;                                        ;N-1;
    }
    printf("\n\nOgrenci bolumu %d donemde bitirir.", donem-1);
    return 0;
}
                                                                                )
```

Video Linki:

https://youtu.be/LvjnFwktLL4