TD 1

# Exo 1

recherche.liste(l,k) O(n)

x = l.tete

TANTQUE x != NULL ET x.clé != k

x = x.succ

Retourner x

inserer.liste(l,k) O(1)

x.secc = l.tete

SI l.tete != NULL

l.tete.pred = x

l.tet = x

x.pred = NULL

supprimer.liste(l,x) O(1)

SI x.pred != NULL

x.pred.succ = x.succ

SINON

l.tete = x.succ

SI x.succ != NULL

x.succ.pred = x.pred

# Exo 2

3.a) O(S+A)

3.b) O(S²)

4.a) O(degré(u))

4.b) O(S)

5.a) O(degré(u))

5.b) O(1)

# Exo 3

2)

cellule.h :

typedef struct cellule\_type{

    int id\_sommet;

    struct cellule\_type \*pred;

    struct cellule\_type \*succ;

}cellule\_t;

int initialiser\_cellule(cellule\_t\* c, int id\_sommet);

cellule.c :

int initialiser\_cellule(cellule\_t\* c, int id\_sommet){

    ...

    return 0;

}

liste.h :

typedef struct{

    cellule\_t\* tete;

}liste\_t;

int initialiser\_liste(liste\_t\* l);

int inserer(liste\_t\* l, cellule\_t\* c);

...

liste.c :

progListe.c :

int main(...){

    cellule\_t cl;

    initialiser\_cellule(&cl,4);

    cellule\_t\* iz;

    iz=(cellule\_t\*) malloc(sizeof(cellule\_t));

    initialiser\_cellule(iz,4);

}

graphe.h :

typedef struct{

    int n\_sommets;

    int oriente;

    int value;

    liste\_t\* l\_adj;

    int \*\* m\_adj;

}graphe\_t;

int initialiser\_graphe(graphe\_t\* g, char \* nom\_fichier);

int initialiser\_graphe(graphe\_t \* g, char \* nom\_fichier){

    FILE \* fichier;

    char ch\_temp[30];

    fichier = fopen(nom\_fichier, "r"); //lecture

    fscanf(fichier, "%s", ch\_temp);

    fscanf(fichier, "%d", g->n\_sommets)

    g->m\_stockage = (int\*) malloc(sizeof(int)\*g->n\_sommets\*g->n\_sommets);

    g->m\_adj = (int\*\*)malloc(sizeof(int\*)\*g->n\_sommets);

    f(i...){

        g->m\_adj[i]=&g->ù\_stockage[i\*g->n\_sommet];

    }

}