

# Alire

---

Ce fichier respecte la norme DokuWiki

Realise par ELHIMDI Yasmine et PARMENTIER Laurent

Groupe 2.

Compte rendu du TP 4.

## Trace d'execution

---

**Remarque:** si vous êtes sur une autre architecture, il faut penser à recompiler la librairie `lifo` pour la pile.

```
cd lib/lifo/  
make distclear  
make tests
```

Doit retourner : OK

---

A l'aide du TP5, on réalisé un partir d'un graphe, la marque topologique du graphe :

```
laubosslink@pc-lbl:/media/files/projects/courses/ensicaen/media/s2/  
algo_avancee/tps/tp6$ ./extra/marquage_topo extra/test_exemple1.txt extra/  
test_exemple1_topo.txt  
Le graphe a ete correctement marqué topologiquement dans le fichier.
```

Ecriture d'un ensemble de test qui permettent de vérifier que sur un exemple donné, (en l'occurrence celui du TP) les valeurs retournées par les fonctions sont bien celles attendues.

- `test_m_adj.c`

```
#include <stdlib.h>  
#include <stdio.h>  
#include <string.h>  
  
#include <lifo.h>  
#include <m_adj.h>
```

```
#define EXEMPLE_GRAPHE_FILE "extra/test_exemple1.txt"
#define EXEMPLE_GRAPHE_TOPO_FILE "extra/test_exemple1_topo.txt"

int main()
{
    int tests = 0;

    m_adj m = m_adj_load_file(EXEMPLE_GRAPHE_FILE);

    /* @todo tester la fonction m_adj_get_duration_from_to(..) */

    /* test des fonctions qui permettent de récupérer une durée d'un sommet
    */

    tests = m_adj_get_duration(m, 1) == 6;

    #if DEBUG_AFFICHE == 1
        printf("duree sommet 1 : %d == 6\n", m_adj_get_duration(m, 1));
    #endif

    tests = m_adj_get_duration(m, 2) == 3;

    #if DEBUG_AFFICHE == 1
        printf("duree sommet 2 : %d == 3\n", m_adj_get_duration(m, 2));
    #endif

    tests = m_adj_get_duration(m, 3) == 6;

    #if DEBUG_AFFICHE == 1
        printf("duree sommet 3 : %d == 6\n", m_adj_get_duration(m, 3));
    #endif

    tests = m_adj_get_duration(m, 4) == 2;

    #if DEBUG_AFFICHE == 1
        printf("duree sommet 4 : %d == 2\n", m_adj_get_duration(m, 4));
    #endif

    tests = m_adj_get_duration(m, 5) == 4;

    #if DEBUG_AFFICHE == 1
        printf("duree sommet 5 : %d == 4\n", m_adj_get_duration(m, 5));
    #endif

    tests = m_adj_get_duration(m, 6) == 3;

    #if DEBUG_AFFICHE == 1
        printf("duree sommet 6 : %d == 3\n", m_adj_get_duration(m, 6));
    #endif

    tests = m_adj_get_duration(m, 0) == 0;
```

```
#if DEBUG_AFFICHE == 1
    printf("duree sommet 0 : %d == 0\n", m_adj_get_duration(m, 0));
#endif

    tests = m_adj_get_duration(m, 7) == 1 && tests;

#if DEBUG_AFFICHE == 1
    printf("duree sommet 7 : %d == 1\n", m_adj_get_duration(m, 7));
#endif

    tests = m_adj_get_duration(m, 8) == 0 && tests;

#if DEBUG_AFFICHE == 1
    printf("duree sommet 8 : %d == 0\n", m_adj_get_duration(m, 8));
    printf("\n");
#endif

/* test de la fonction qui permet de récupérer les sommet prédécesseur
d'un sommet */

    Lifo l = m_adj_pred(m, 6);
    int *elmt;

#if DEBUG_AFFICHE == 1
    printf("predecesseur de 6 : {");
#endif

    if(lifo_is_empty(l))
        tests = 0;

    while(!lifo_is_empty(l))
    {
        elmt = (int *) lifo_pop_elmt(l);

#if DEBUG_AFFICHE == 1
        printf("%d, ", *elmt);
#endif

        /* si on ne trouve pas le sommet 1 ou 4 */
        if(*elmt != 1 && *elmt != 4)
        {
            tests = 0;
        }
    }

#if DEBUG_AFFICHE == 1
    printf("} <=> {1, 4}\n");
#endif

    l = m_adj_pred(m, 1);
```

```
#if DEBUG_AFFICHE == 1
    printf("predecesseur de 1 : {");
#endif

    if(lifo_is_empty(l))
        tests = 0;

    while(!lifo_is_empty(l))
    {
        elmt = (int *) lifo_pop_elmt(l);

#if DEBUG_AFFICHE == 1
        printf("%d, ", *elmt);
#endif

        /* si on ne trouve pas le sommet 0*/
        if(*elmt != 0)
        {
            tests = 0;
        }
    }

#if DEBUG_AFFICHE == 1
    printf("} <=> {0}\n");
    printf("\n");
#endif

/* test que les noms de tache renvoie correspondent à celle du sommet
*/
    tests = strcmp(m_adj_get_tache(EXEMPLE_GRAPHE_FILE, 1), "A") == 0
    && tests;

#if DEBUG_AFFICHE == 1
    printf("tache du sommet 1 : %s == A\n", m_adj_get_tache(
    EXEMPLE_GRAPHE_FILE, 1));
#endif

    tests = strcmp(m_adj_get_tache(EXEMPLE_GRAPHE_FILE, 4), "D") == 0
    && tests;

#if DEBUG_AFFICHE == 1
    printf("tache du sommet 4 : %s == D\n", m_adj_get_tache(
    EXEMPLE_GRAPHE_FILE, 4));
#endif

    tests = strcmp(m_adj_get_tache(EXEMPLE_GRAPHE_FILE, 8), "Omega") ==
    0 && tests;

#if DEBUG_AFFICHE == 1
    printf("tache du sommet 8 : %s == Omega\n", m_adj_get_tache(
```

```
EXEMPLE_GRAPHE_FILE, 8));
#endif
    tests = strcmp(m_adj_get_tache(EXEMPLE_GRAPHE_FILE, 6), "F") == 0
&& tests;

#if DEBUG_AFFICHE == 1
    printf("tache du sommet 6 : %s == F\n", m_adj_get_tache(
EXEMPLE_GRAPHE_FILE, 6));
    printf("\n");
#endif

/* test de la fonction m_adj_get_sommet */

    tests = m_adj_get_sommet(EXEMPLE_GRAPHE_FILE, "A") == 1 && tests;

#if DEBUG_AFFICHE == 1
    printf("numero du sommet A : %d == 1\n", m_adj_get_sommet(
EXEMPLE_GRAPHE_FILE, "A"));
#endif

    tests = m_adj_get_sommet(EXEMPLE_GRAPHE_FILE, "D") == 4 && tests;

#if DEBUG_AFFICHE == 1
    printf("numero du sommet D : %d == 4\n", m_adj_get_sommet(
EXEMPLE_GRAPHE_FILE, "D"));
#endif

    tests = m_adj_get_sommet(EXEMPLE_GRAPHE_FILE, "Omega") == 8 &&
tests;

#if DEBUG_AFFICHE == 1
    printf("numero du sommet Omega : %d == 8\n", m_adj_get_sommet(
EXEMPLE_GRAPHE_FILE, "Omega"));
#endif

    tests = m_adj_get_sommet(EXEMPLE_GRAPHE_FILE, "F") == 6 && tests;

#if DEBUG_AFFICHE == 1
    printf("numero du sommet F : %d == 6\n", m_adj_get_sommet(
EXEMPLE_GRAPHE_FILE, "F"));
    printf("\n");
#endif

/* test de la fonction m_adj_dtot */

    tests = m_adj_dtot(m, m_adj_get_sommet(EXEMPLE_GRAPHE_FILE, "G"))
== 9 && tests;

    tests = m_adj_dtot(m, m_adj_get_sommet(EXEMPLE_GRAPHE_FILE, "F"))
```

```
== 6 && tests;

    tests = m_adj_dtot(m, m_adj_get_sommet(EXEMPLE_GRAPHE_FILE, "D"))
== 3 && tests;

    tests = m_adj_dtot(m, m_adj_get_sommet(EXEMPLE_GRAPHE_FILE, "E"))
== 3 && tests;

#if DEBUG_AFFICHE == 1
    /* @todo les tests de dtot */
#endif

#if DEBUG_AFFICHE == 1
    m_adj_date_plus_tot(EXEMPLE_GRAPHE_FILE, EXEMPLE_GRAPHE_TOPO_FILE,
m);
    printf("\n");
#endif

/* tests de m_adj_dtard*/

    tests = m_adj_dtard(m, m_adj_get_sommet(EXEMPLE_GRAPHE_FILE, "Omega"))
== 10 && tests;

#if DEBUG_AFFICHE == 1
    printf("date plus tard Omega : %d == 10\n", m_adj_dtard(m,
m_adj_get_sommet(EXEMPLE_GRAPHE_FILE, "Omega")));
#endif

    tests = m_adj_dtard(m, m_adj_get_sommet(EXEMPLE_GRAPHE_FILE, "G"))
== 9 && tests;

#if DEBUG_AFFICHE == 1
    printf("date plus tard G : %d == 9\n", m_adj_dtard(m,
m_adj_get_sommet(EXEMPLE_GRAPHE_FILE, "G")));
#endif

    tests = m_adj_dtard(m, m_adj_get_sommet(EXEMPLE_GRAPHE_FILE, "F"))
== 6 && tests;

#if DEBUG_AFFICHE == 1
    printf("date plus tard F : %d == 6\n", m_adj_dtard(m,
m_adj_get_sommet(EXEMPLE_GRAPHE_FILE, "F")));
#endif

    tests = m_adj_dtard(m, m_adj_get_sommet(EXEMPLE_GRAPHE_FILE, "D"))
== 4 && tests;

#if DEBUG_AFFICHE == 1
    printf("date plus tard D : %d == 4\n", m_adj_dtard(m,
m_adj_get_sommet(EXEMPLE_GRAPHE_FILE, "D")));
#endif
```

```

    tests = m_adj_dtard(m, m_adj_get_sommet(EXEMPLE_GRAPHE_FILE, "E"))
    == 5 && tests;

#if DEBUG_AFFICHE == 1
    printf("date plus tard E : %d == 5\n", m_adj_dtard(m,
m_adj_get_sommet(EXEMPLE_GRAPHE_FILE, "E")));
    printf("\n");
#endif

#if DEBUG_AFFICHE == 1
    m_adj_date_plus_tard(EXEMPLE_GRAPHE_FILE, EXEMPLE_GRAPHE_TOPO_FILE,
m);
    printf("\n");
#endif

    if(tests == 1)
        printf("m_adj: \033[32mOK\033[0m\n");
    else
        printf("m_adj: \033[31mproblemes durant les tests\033[0m\n");

    return 0;
}

```

Code retourné :

```

laubosslink@pc-lbl:/media/files/projects/courses/ensicaen/media/s2/
algo_avancee/tps/tp6$ make tests
gcc -Wall -Wextra -ansi -pedantic -std=c99 -I ./inc -I ./lib/*/inc
-DDEBUG_AFFICHE=1 src/test_m_adj.c obj/m_adj.o lib/lifo/obj/lifo.o -o bin/
debug/test_m_adj
./bin/debug/test_m_adj
duree sommet 1 : 6 == 6
duree sommet 2 : 3 == 3
duree sommet 3 : 6 == 6
duree sommet 4 : 2 == 2
duree sommet 5 : 4 == 4
duree sommet 6 : 3 == 3
duree sommet 0 : 0 == 0
duree sommet 7 : 1 == 1
duree sommet 8 : 0 == 0

predecesseur de 6 : {4, 1, } <=> {1, 4}
predecesseur de 1 : {0, } <=> {0}

tache du sommet 1 : A == A
tache du sommet 4 : D == D
tache du sommet 8 : Omega == Omega
tache du sommet 6 : F == F

```

```
numero du sommet A : 1 == 1
numero du sommet D : 4 == 4
numero du sommet Omega : 8 == 8
numero du sommet F : 6 == 6
```

La tâche Alpha commence au plus tot la semaine 0 et sera donc finit au bout de 0 semaines.

La tâche C commence au plus tot la semaine 0 et sera donc finit au bout de 6 semaines.

La tâche B commence au plus tot la semaine 0 et sera donc finit au bout de 3 semaines.

La tâche A commence au plus tot la semaine 0 et sera donc finit au bout de 6 semaines.

La tâche E commence au plus tot la semaine 3 et sera donc finit au bout de 7 semaines.

La tâche D commence au plus tot la semaine 3 et sera donc finit au bout de 5 semaines.

La tâche F commence au plus tot la semaine 6 et sera donc finit au bout de 9 semaines.

La tâche G commence au plus tot la semaine 9 et sera donc finit au bout de 10 semaines.

La tâche Omega commence au plus tot la semaine 10 et sera donc finit au bout de 10 semaines.

```
date plus tard Omega : 10 == 10
```

```
date plus tard G : 9 == 9
```

```
date plus tard F : 6 == 6
```

```
date plus tard D : 4 == 4
```

```
date plus tard E : 5 == 5
```

La tâche Alpha doit commencer au plus tard la semaine 0.

La tâche C doit commencer au plus tard la semaine 3.

La tâche B doit commencer au plus tard la semaine 1.

La tâche A doit commencer au plus tard la semaine 0.

La tâche E doit commencer au plus tard la semaine 5.

La tâche D doit commencer au plus tard la semaine 4.

La tâche F doit commencer au plus tard la semaine 6.

La tâche G doit commencer au plus tard la semaine 9.

La tâche Omega doit commencer au plus tard la semaine 10.

m\_adj: OK

## Difficultés

La difficulté du TP a essentiellement été dans l'écriture des fonctions permettant de réaliser dtot, et dtard.



## Bogues

---

Aucun bogue connu pour le moment

## A améliorer

---

- Améliorer le nom des fonctions
- Implémenter l'ajout du sommet Omega, Alpha via le code m\_adj.c
- La possibilité de lire les fichiers sous différents formats (commentaires au début)

From:

<https://ensicaen.singular.society-lbl.com/> - **Ensicaen**

Permanent link:

[https://ensicaen.singular.society-lbl.com/s2:algo\\_avancee:tp6:start](https://ensicaen.singular.society-lbl.com/s2:algo_avancee:tp6:start)

Last update: **2014/05/31 16:50**

