

Objectifs :

- Construire la liste d'adjacence à partir d'un fichier
- Construire un graphe à partir de la matrice d'adjacence

On considère le graphe suivant:

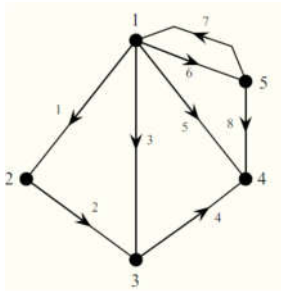


Figure 1

La liste d'adjacence se rapportant à la figure 1 sera représentée par la liste chaînée suivante :

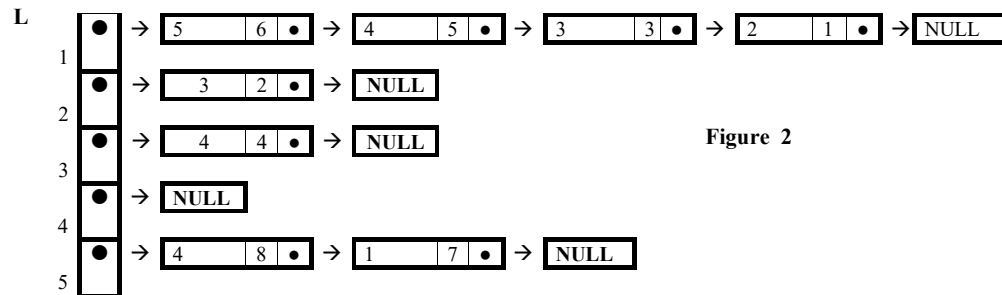


Figure 2

1) Définition des types de base pour la construction de la liste d'adjacence :

Pour construire la liste d'adjacence, associée à un graphe, on définit les structures suivantes :

a) Définition de la structure de la liste chaînée

```
typedef struct sommet{
  int noeud;
  float poids;
  struct sommet *suiv;
} sommet;
```

b) Définition d'un type

```
typedef struct sommet* liste ; // déclaration d'un type pointeur sur sommet
```

2) Travail demandé :

2.1) Construction de la liste d'adjacence à partir d'un fichier

Pour la construction on a besoin des fonctions suivantes :

- a) **liste ajout_arc(liste l, int s, float v)** : permet d'ajouter un nœud (*num et poids*) en tête de liste.
- b) **liste * remp_liste_adj(FILE * f, int *nb)** : La fonction **remp_liste_adj** retourne l'adresse du tableau des listes chaînées des différents sommets du graphe et le nombre de sommet.
- c) **void affiche_liste(liste *L, int nb)** : permet d'afficher chaque sommet avec ses successeurs à partir de la liste d'adjacence.

2.2) Construction de la matrice des successeurs à partir de la liste d'adjacence

- a) **void Rzero(int M[Nbs][Nbs], int s)** : permet de mettre à zéro une matrice carrée **M** d'ordre **s**. (voir tp2)
- b) **void affichmat(int M[Nbs][Nbs], int s)** : permet d'afficher une matrice carrée **M**. (voir tp2)
- c) **void Lsucc(liste *L, int Msucc[Nbs][Nbs], int s)** : permet de remplir la matrice de successeurs **Msucc** à partir de la liste d'adjacence.
- d) **void Lpred(liste *L, int Mpred[Nbs][Nbs], int s)** : permet de remplir de la matrice des prédécesseurs **Mpred** à partir de la liste d'adjacence.

2.3) Construction de la matrice d'adjacence à partir du Msucc

void Matadj(int Msucc[Nbs][Nbs], int Madj[Nbs][Nbs], int s) : permet de remplir la matrice d'adjacence **Madj** à partir de la matrice des successeurs **Msucc**.

2.4) Construction de la liste d'adjacence à partir de la matrice d'adjacence

liste * lectmat(int Madj[nmax][nmax], int s) permet de construire la liste d'adjacence à partir de la matrice d'adjacence **Madj** de dimension **s**, on suppose que la valeur du poids est égale à 0.