Travaux Pratiques d'algorithmique n°4

Cours d'Informatique de Deuxième Année

—DEUG Sciences S4-MI—

Problèmes d'ordonnancement

La mise en oeuvre d'une recette de cuisine complexe conduit à des problèmes d'ordonnancement dans lesquels un ensemble de tâches doivent être exécutés dans un certain ordre. Par exemple, éplucher les pommes de terre, les couper en cubes sont deux tâches qui doivent impérativement être exécutées de manière séquentielle alors que lécher la casserolle pendant que le gâteau cuit sont deux tâches pouvant être exécutées en parallèle.

On peut représenter ces problèmes d'ordonnancement sous la forme d'un graphe où chaque sommet représente une tâche et chaque arête une relation de succession directe : il existe un arc de i vers j si i doit être exécutée avant j.

On considère les déclarations suivantes où un graphe est représenté par la structure classique :

```
struct graphe
{
  int FS[MAX];
  int APS[MAX];
  int nbsommets;
};

typedef struct graphe Graphe;     /* nouveau type : Graphe              */
```

▸ **Exercice 1.** ***(Fonctions de base)***

Réutiliser ou ré-écrire les fonctions élémentaires permettant de :

1. *créer les tableaux FS et APS d'un graphe à partir d'un fichier ayant le format suivant :*

   ```
   2 4 3 5 0 3 0 6 0 6 0 4 0 0
   ```

2. *afficher la liste des successeurs de chaque sommet du graphe.*

▸ **Exercice 2.** ***(Algorithme du rang)*** *En supposant que les tâches ont toutes la même durée, écrire une fonction permettant de calculer le temps minimal de préparation de la recette.*

▸ **Exercice 3.** ***(Détection des cycles)***

- *Modifier la fonction précédente pour intégrer la détection de cycles éventuels dans l'élaboration d'une recette.*
- *Utiliser un algorithme de descente prioritairement en profondeur pour réaliser la même opération de détection.*