**Algorithme de Navigation dans un graphe orienté.**

L’objet de cet algorithme est de permettre la représentation d’un graphe orienté cyclique.

Les nœuds du graphe références des « étapes », le graphe en lui-même définit une activité composée d’étapes. Dans l’exemple ci-dessous, nous voyons les différentes retapes d’une activité numérotées de 1 à 7, avec des transitions numérotées. Les étapes mises en exergue en rouge représentent un cycle dans le graphe.



Notre Interface de navigation doit permettre certaines fonctionnalités tout en garantissant l’intégrité des données. Ces fonctionnalités sont :

* La suppression
* L’insertion
* Les succeseurs
* Les predecesseurs
* Chemins.

Contraintes :

Données savable.

Ajouter une étape a tout moment.

Sauvegarder les étapes parcourues.

Approche Matricielle.

Avec une bonne nomenclature des transitions du graphe, une représentation matricielle semble sied à cette problématique. Cependant, une représentation traditionnelle peut aussi fonctionner, mais est peu pratique durant la sauvegarde.

Idée de base :

Pour un graphe de N nœuds, nous déclarons une matrice dynamique carrée NxN. Les nœuds constituent les index des lignes est des colonnes, alors que les cellules numérotent les transitions.

Exemple pour la transition de 1 à 2, nous mettrons une valeur « level » à la cellule (1,2).

La valeur du level est le max +1 des transitions entrantes dans l’état 1.

Le graphe précédent aura comme représentation matricielle :

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0 |  | 1 |  |  |  |  |  |  |  |
| 1 |  |  | 2 | 2 |  | 2 |  | 2 |  |
| 2 |  |  |  |  | 3 |  |  |  |  |
| 3 |  |  |  | **3** |  | 3 |  | 3 |  |
| 4 |  |  |  |  |  |  |  | 4 |  |
| 5 |  |  |  |  |  |  | 4 |  |  |
| 6 |  |  |  |  |  |  |  | 5 |  |
| 7 |  |  |  | 6 |  | 6 |  | 6 | 6 |
| 8 |  |  |  |  |  |  |  |  |  |