

## Séance 05 - Exercices à faire à la maison

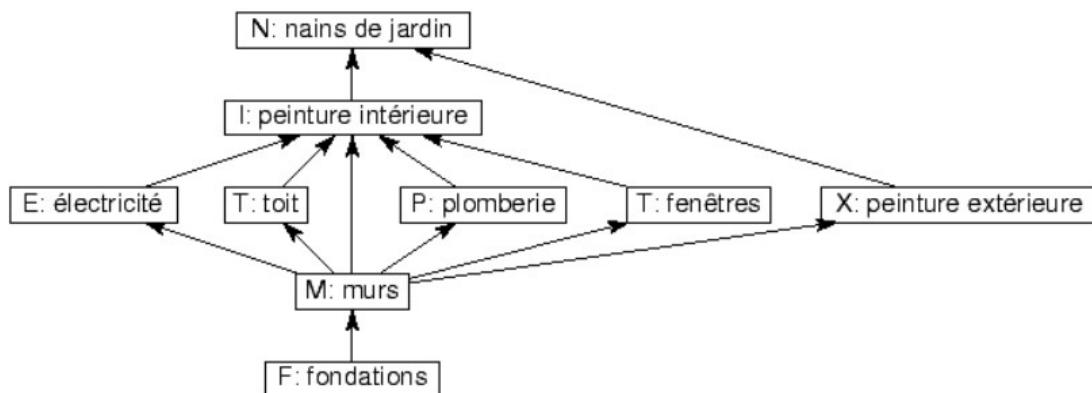
**Note sans rapport avec l'exercice ci-dessous :**

Si vous souhaitez vous entraîner à exécuter l'algorithme de parcours en profondeur, je vous conseille le site suivant : <https://opendsa-server.cs.vt.edu/embed/DFSPE>

## exercice 1

On veut construire une maison, ce qui consiste en 9 tâches, le plus rapidement possible, avec les contraintes suivantes:

- Certaines tâches dépendent d'autres tâches (un arc (A,B) indique que la tâche A doit être exécutée avant la tâche B)
  - Toutes les tâches demandent une semaine de travail.
  - Chaque ouvrier ne peut travailler que sur une tâche par semaine
  - Il n'y a pas de gain de temps si plusieurs ouvriers travaillent sur la même tâche



- 1) Organisez un emploi du temps pour un ouvrier, de façon à construire la maison le plus rapidement possible.

- 2) Quelle propriété doit vérifier le graphe pour que la maison puisse être construite ?
  - 3) Quel est le rapport entre ce problème et les sujets étudiés dans le TD 5 ? En déduire un algorithme qui détermine un emploi du temps valide pour un ouvrier.
  - 4) Organisez un emploi du temps pour construire la maison le plus vite possible, cette fois-ci pour deux ouvriers :

Ouvrier 2									
-----------	--	--	--	--	--	--	--	--	--

- 5) Même question pour quatre ouvriers :

Semaine	1	2	3	4	5	6	7	8	9
Ouvrier 1									
Ouvrier 2									
Ouvrier 3									
Ouvrier 4									

- 6) Et s'il y a plus d'ouvriers ?

*Dans le jargon de la recherche opérationnelle, nous avons **ordonnancé des tâches** (construction des fondations, ...) sur plusieurs **processeurs** (les ouvriers), avec des **contraintes de précédences** (un ordre partiel).*

*Un **ordonnancement** est optimal s'il minimise la **fonction objectif** (ici la durée totale).*

- 7) Cherchez un algorithme pour ordonner optimalement des tâches avec contraintes de précédences sur un nombre infini de processeurs.  
 Quelle est la complexité de cet algorithme?  
 Démontrez l'optimalité de l'ordonnancement obtenu.

**Indice :** Il s'agit d'un algorithme glouton. À chaque étape, on maximise localement le nombre de tâches effectuées, et il se trouve que le problème est suffisamment simple pour que le résultat soit un optimal global.