

# Asie– 2024 – sujet1 - Correction

## Exercice 2 (6 points)

---

### 1. Nature du graphe

C'est un **graphe orienté**.

👉 *Commentaire : Les arcs représentent des dépendances (« doit être fait avant »). Le sens des flèches est donc essentiel.*

---

### 2. Ordres possibles ou non

- (f) puis (g) → **Oui**
- (g) puis (f) → **Non**
- (i) puis (j) → **Oui**
- (j) puis (i) → **Non**

👉 *Commentaire : Lorsqu'un arc va de X vers Y, X doit être réalisé avant Y. L'ordre inverse est donc impossible.*

---

### 3. Tâches nécessaires avant (k)

Il faut toutes les tâches permettant d'obtenir la pâte prête avant cuisson, notamment :

- préparation farine
- préparation eau
- préparation sel
- préparation levure
- faire tiédir l'eau
- délayer levure
- laisser reposer
- verser farine
- verser sel
- mélanger
- verser eau+levure
- pétrir

👉 *Commentaire : On remonte toutes les dépendances en amont de (k). Il faut uniquement celles reliées par un chemin orienté vers (k).*

---

#### 4. Cycle ?

Le graphe **ne contient pas de cycle**.

👉 *Commentaire : Une recette ne peut pas contenir de dépendance circulaire (sinon on ne pourrait jamais commencer).*

---

#### 5. Ordre possible – Figure 2

Un ordre valide est par exemple :

0 → 1 → 2 → 3 → 4 → 5

(ou toute autre permutation respectant les dépendances)

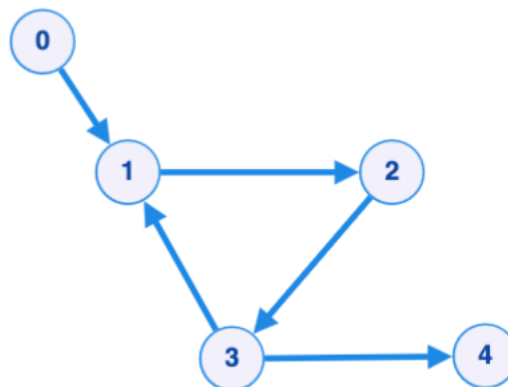
👉 *Commentaire : On réalise un tri topologique : on choisit toujours un sommet sans prédécesseur restant.*

---

#### 6. Graphe associé à la matrice M

M :

0 → 1  
1 → 2  
2 → 3  
3 → 1  
3 → 4



👉 *Commentaire :*

- *Il existe un cycle : 1 → 2 → 3 → 1*
  - *4 n'a pas de successeur.*
- 

#### 7. Existe-t-il un ordre ?

**Non.** *Le cycle 1 → 2 → 3 → 1 empêche tout ordre de réalisation. Un tri topologique est impossible en présence d'un cycle.*

---

## 8. Fonction mystere

### Valeur de ok

ok = False

👉 *Commentaire : La fonction détecte un cycle via les tableaux ouverts et fermes.*

---

### Déroulement des appels

Initialisation :

```
ouverts = [F, F, F, F, F]
fermes   = [F, F, F, F, F]
```

Appel 1 :

```
mystere(M, 1, ...)
ouverts = [F, T, F, F, F]
```

Appel 2 :

```
mystere(M, 2, ...)
ouverts = [F, T, T, F, F]
```

Appel 3 :

```
mystere(M, 3, ...)
ouverts = [F, T, T, T, F]
```

Appel 4 :

```
mystere(M, 1, ...)
```

Or 1 est déjà ouvert → cycle détecté.

👉 *Commentaire : Un sommet déjà « ouvert » signifie qu'on revient dans la branche courante → présence d'un cycle.*

---

## 9. Cas général où mystere renvoie False

La fonction renvoie **False** lorsqu'elle détecte un cycle.

👉 **Commentaire :** Plus précisément, lorsqu'un sommet est revisité alors qu'il est déjà marqué comme « ouvert ».

---

## 10. Structure de pile

Instructions :

```
essai = Pile()
essai.empiler(3)
essai.empiler(2)
essai.empiler(10)
elt = essai.depiler()
elt = essai.depiler()
```

Valeur finale de `elt` : 2

👉 **Commentaire :**

*Une pile est LIFO (Last In First Out).  
Ordre dans la pile : 3, 2, 10  
On retire 10 puis 2.*

---

## 11. Compléter mystere pour ordre\_realisation

À la ligne 24 il faut empiler le sommet lorsqu'il est totalement traité :

```
pile.empiler(sommet)
```

👉 **Commentaire :** On empile le sommet **après avoir traité tous ses successeurs**. Cela correspond au principe du tri topologique par parcours en profondeur.

---