

S2.02 : Rapport

DAG :

La classe DAGVerifier fournit des méthodes permettant de vérifier si un graphe orienté est un DAG en détectant la présence de cycles par une exploration en profondeur (DFS).

estDAG(...) ;

- Renvoi true si le graphe est un DAG
- Fonctionnement : Parcourt tous les nœuds du graphe et pour chaque nœud non encore visité, lance la méthode aUnCycle.

aunCycle(...) ;

- Fonction récursive qui renvoi true si un cycle existe à partir du nœud courant
- Fonctionnement : Marque courant comme visité et l'ajoute à pileRec. Puis explore récursivement ses voisins et si un voisin est déjà dans pileRec, un cycle est détecté.

estDansListe(...) ;

- Vérifie si une valeur est dans la liste

supprimerDeListe(...) ;

- Supprime un élément de la liste s'il y est.
- Fonctionnement : Recherche la dernière position de l'élément et le supprime.

Exhaustive (Backtracking):

Trouver l'affectation optimale en affectant à chaque DPS un secouriste qui possède toutes les compétences requises, et en veillant à ce que chaque secouriste ne soit affecté qu'à un seul DPS.

affecterExhaustif(...) ;

- lancer la recherche exhaustive pour renvoyer la meilleur affectation possible.
- Fonctionnement : Lance la fonction récursive essayerToutesLesAffectations(...).

essayerToutesLesAffectations(...) ;

- explorer toutes les combinaisons possibles d'affectation DPS → secouriste, de manière récursive.
- Fonctionnement : À chaque appel, on traite un DPS, on essaie de l'affecter à chaque secouriste non affecté, à condition que ses compétences correspondent aux besoins du DPS. Pour chaque affectation valide, on l'ajoute à la solution courante et on appelle récursivement la méthode sur le DPS suivant.

Glouton :

Affecter le plus de DPS possible en leur assignant un secouriste qui possède toutes les compétences nécessaires, sans réutiliser de secouriste. Cette méthode ne garantit pas une solution optimale, mais elle est beaucoup plus rapide.

affecterGlouton(...);

- Fonctionnement : Initialise une map affectations et une liste secouristesDejaAffectes pour suivre les secouristes utilisés.

Pour chaque DPS :

- Récupère ses besoins en compétences.
- Parcourt tous les secouristes :
 - ◆ Ignore ceux déjà affectés.
 - ◆ Si un secouriste libre possède toutes les compétences nécessaires alors il est affecté au DPS, il est ajouté à la liste des secouristes déjà affectés et on passe au DPS suivant.

```
Méthode exhaustive :  
DPS : 4 → Secouriste : 5  
DPS : 2 → Secouriste : 1  
DPS : 1 → Secouriste : 2  
DPS : 6 → Secouriste : 6  
DPS : 8 → Secouriste : 7  
DPS : 9 → Secouriste : 8  
DPS : 3 → Secouriste : 3  
DPS : 5 → Secouriste : 4  
Nombre de DPS couverts : 8  
Temps (ms) : 19.431449
```

```
Méthode gloutonne :  
DPS : 4 → Secouriste : 5  
DPS : 2 → Secouriste : 1  
DPS : 1 → Secouriste : 2  
DPS : 6 → Secouriste : 6  
DPS : 8 → Secouriste : 7  
DPS : 9 → Secouriste : 8  
DPS : 3 → Secouriste : 3  
DPS : 5 → Secouriste : 4  
Nombre de DPS couverts : 8  
Temps (ms) : 0.856689
```

Critère	Exhaustive	Glouton
Temps d'exécution	Lente	Rapide
Mémoire utilisée	Élevée	Faible
Solution optimale garantie	Oui	Non
Complexité du code	Plus complexe	Très simple