Former des groupes de travail dans votre classe de NSI

1 Situation

La classe de première NSI est composée de 27 élèves qui doivent réaliser un projet informatique par groupes de 3.
Vous devez réaliser pour le professeur, qui est votre ńclientż, un programme qui fasse une proposition de la composition de ces groupes de 3 élèves, et qui leur convienne le mieux possible .

Pour vous aider, le professeur a réalisé une petite enquête : il a demandé à chaque élève de citer :

- un élève avec qui il voudrait absolument travailler (affecté de la valeur +2),
- un avec qui il voudrait travailler (affecté de la valeur +1)
- et un avec qui il ne voudrait pas travailler (affecté de la valeur -1)

Ces données vont être fournies par le professeur sous la forme dun dictionnaire : Par exemple :

```
Classe_NSI={ 'Alexia': {'Kellian': 2, 'Christelle': 1, 'Alice': -1}, etc.}
```

Ici Alexia voudrait absolument travailler avec Kellian, voudrait bien travailler avec Christelle, et pas du tout avec Alice.

2 Cahier des charges

- **Version obligatoire**
  - On ne dispose que des élèves avec qui chacun veut travailler :
    `Par exemple : Classe_NSI={ 'Alexia': {'Kellian': 2, 'Christelle': 1}, etc.}`.
  - Ce dictionnaire sera dans le fichier data _exemple.json
  - Le professeur fournit la fonction def lire-entree _json qui retourne le dictionnaire.
  - On ne tient pas compte des préférences (par exemple celle pour Kellian par rapport à Christelle).
- **Version luxe**
  Tenir compte de tous les renseignements donnés (+2, +1 ou -1)

Dans tous les cas, vous devrez travailler par groupes de deux élèves :

1. Réaliser en Python le programme demandé, avec des tests pour les principales fonction.
2. Faire un exposé avec présentation, et démonstration de l'exécution du programme.
3. Rédiger un dossier contenant (au moins 10 pages hors annexe) :
   - le cahier des charges

- un planning des différentes tâches exécutées par chacun
- une explication des problèmes rencontrés
- des copies d'écran des résultats obtenus
- le code complet du programme en annexe

3 Planning

1. Phase 0 : Mise en place de framagit (0,5 semaine)

2. Phase 1 : (2 semaines)
   - 1er lot : Représentation des graphes :

     Créer un algorithme, qui crée un dictionnaire substituant des numéros aux prénoms, ne gardant que les renseignements nécessaires, (pas de +2 ou +1) pour que les graphes soient plus lisible.
     Faire une dictionnaire de correspondance numéro/prénom

     Écrire une fonction qui crée un graphe orienté à partir dun dictionnaire

     Représenter un graphe donné par un dictionnaire, et créer un fichier png avec ce graphique (utiliser les bibliothèques matplotlib, networkx, numpy)

   - 2e lot : Travail sur les chemins :
     On commence par travailler sans se préoccuper des choix +2 ou +1.
     Créer des fonctions, qui à partir dun dictionnaire du type {0 :[20,15],1 :[18,13], ...} permettent de déterminer tous les cycles de longueur 3. Vous pouvez par exemple commencer par chercher tous les chemins de longueur 3.

     Créer un graphe réduit dans lequel tous les cycles de longueur 3 ont été supprimés.

3. Phase 2 : (1 semaine)

   Réunion des porteurs du projet : mise en commun du travail réalisé et intégration des différentes fonctions. Représenter les graphes réduit créés dans le 2e lot. Vous réaliserez ce travail à 2, en classe.

   Point d'étape avec votre client (le professeur).

4. Phase 3 : (2 semaines)
   - 1er lot : travail sur la matrice dadjacence

     Créer une fonction qui à partir dun dictionnaire du type {0 :[20,15],1 :[18,13],... } retourne la matrice dadjacence.
   - 2e lot :

     Créer une fonction qui retourne la liste des élèves qui ne sont demandés par personne

     Créer une fonction qui retourne la liste des binômes : Par exemple si Alexia(5) veut travailler avec Kellian(11), et si Kellian(11) veut travailler avec Alexia(5) cette fonction retournera entre autres [5,11]. On propose de donner la liste par ordre croissant.

5. Phase 4 : (1 semaine)

   Réunion des porteurs du projet : mise en commun, aide mutuelle.

   Détermination des lots restant à réaliser pour finaliser le projet. Répartition du travail entre vous
   Point d'étape avec votre client (le professeur).

6. Phase 5 : (2 semaines)
   Chacun réalise le lot décidé en phase 4.
   Point d'étape hebdomadaire avec votre client (le professeur).

7. Phase 6 : (2 semaines)
   Mise en commun du travail, intégration des différentes fonctions dans le programme.
   Point d'étape hebdomadaire avec votre client (le professeur).
   Rectifications éventuelles

8. Phase 7 : (1 semaine)
   Préparation de l'exposé et de tous les supports demandés (diaporama, dossier...)

4 Évaluation du projet

Les points suivants seront évalués :

- Le programme fonctionne et répond au cahier des charges
- engagement personnel : chaque élève évalue de façon confidentielle auprès du professeur l'implication de son camarade dans le projet.
- respect des dates du planning
- qualité du programme créé (structure générale, présence de tests, de commentaires etc...)
- qualité de la prestation orale lors de l'exposé
- qualité des supports (diaporama, dossier)

4.1 Grille d'évaluation

Critère	Niveau (MI, MF, BM, TBM)
Respect du cahier des charges	
Le code est correctement exécutable	
Les tests sont respectés et validés	
Les fonctions sont documentées	
Toutes les fonctions sont testées	
Les problèmes sont identifiés par des tickets	
Les dates des jalons ont été respectées	
Avis des camarades sur l'implication	
Avis de l'enseignant sur l'implication	

5 Quelques ressources documentaires sur le sujet

1. Un problème complexe :
   `https://fr.wikipedia.org/wiki/Partitionnement_de_graphe`
   `https://en.wikipedia.org/wiki/Graph_partition`

2. Des solutions algorithmiques :
   `https://en.wikipedia.org/wiki/Cycle_detection`

https://stackoverflow.com/questions/10185773/graph-theory-splitting-a-graph
https://cstheory.stackexchange.com/questions/16387/k-clustering-of-a-graph-maximizing-intra-cluster-weights

3. ou pratiques :
https://stackoverflow.com/questions/53379575/networkx-splitting-a-graph-into-n-subgraphs-based-on-edge-weight
https://framagit.org/ketluts/gestiondeclasses/blob/master/app%2Fstudent%2Fviews%2Fsociogramme.php