Lycée Hoche 2024-2025
ECG1 A Mathématiques

Programme de colle n^o 20 : Théorie des graphes, chapitre complet.

*Semaine du lundi **10 mars**.*

Le programme de la semaine précédente est toujours au programme de cette semaine.

Généralités sur les graphes

20.1 Graphes non orientés : définition, vocabulaires : sommets adjacents, sommet isolé, boucle, graphe simple, ordre d'un graphe, extrémités d'une arête, degré d'un sommet.

Les graphes considérés dans ce cours sont sans multi-arêtes.

20.2 Le lemme des poignées de mains pour les graphes non orientés. Exemples d'application. Corollaire : tout graphe non orienté a un nombre pair de sommets de degré impair.

20.3 Chaîne d'un graphe non orienté. Sommets reliés par une chaîne. Graphes connexes.

Une chaîne de longueur 0 *est simplement donnée par un sommet.*

20.4 Graphes orientés : définition, ordre, boucle, graphes simples, origine/extrémité initiale/extrémité sortante d'une arête, but/extrémité finale/extrémité entrante d'une arête, degré entrant, sortant ou total d'un sommet.

20.5 Lemme des poignées de mains pour les graphes orientés (deux versions : avec les degrés entrants et sortants ou avec les degrés totaux).

20.6 Chemin d'un graphe orienté. Chemin d'un sommet s vers un sommet s'. Graphes orientés connexes.

L'orientation des arêtes compte pour les chemins. La connexité au programme pour les graphes orientés est la connexité forte (existence d'un chemin de s vers s' pour tous sommets s et s').

Matrice d'adjacence et connexité

20.7 Matrice d'adjacence d'un graphe (associée à une numérotation de ses sommets, dans le cas orienté ou non). Caractère symétrique de la (de toute) matrice d'adjacence d'un graphe non orienté. Degrés et somme des coefficients d'une même ligne (ou colonne) de "la" matrice d'adjacence (cas orienté ou non).

20.8 Si M est la matrice d'adjacence d'un graphe G orienté (resp. non orienté) de sommets numérotés $s_1, ..., s_n$, alors le coefficient d'indice (i, j) de M^k est le nombre de chemins (resp. de chaînes) de longueur k du sommet s_i vers le sommet s_j (pour tous i, j, k convenables).

20.9 Un graphe G (orienté ou non) d'ordre n et de matrice d'adjacence M est connexe si et seulement si $I_n + M + ... + M^{n-1}$ est une matrice à coefficients strictement positifs. Démonstration utilisant le lemme suivant : s'il existe un chemin (ou une chaîne) d'un sommet s vers un sommet s', alors il existe un tel chemin de longueur au plus $n - 1$.

Quelques thèmes supplémentaires classiques

20.10 Exemple d'utilisation de récurrence pour démontrer des résultats généraux.

Rapidement abordés en cours, les élèves ont un sujet corrigé pour comprendre ces démarches.

20.11 Graphe (orientés ou non) pondérés : définition et idée.

Cette notion sera abordée plus largement en TP.

20.12 Notion de graphe complet. Calcul des puissances de "la" matrice d'adjacence M d'un graphe complet.

20.13 Notion de chaîne fermée, de cycle, de cycle eulérien et de graphe eulérien (cas non orienté seulement). Un graphe connexe est eulérien si et seulement si tous ses sommets sont de degré pair.

Sans démonstration, mais l'idée a été expliquée sur un exemple.

20.14 Notion de graphes bipartis. Notion de sous-graphe, de sous-graphe induit et de composante connexe.

La notion de composante connexe est HP : les exercices qui en ont besoin doivent la redéfinir...

20.15 Annexe : un problème sur les graphes sans triangles (théorème de Mantel)

Sujet corrigé à faire en autonomie.

Python : le type `array`

20.16 Effet des opérateurs de comparaison sur les variables de type `array`.

Si `M` est un objet de type `array` à coefficients booléens, les commandes `M.all()` et `M.any()` ont été brièvement abordées à l'oral et peuvent être utilisées si la question ne l'interdit pas.

20.17 Matrices : commandes donnant des matrices remarquables (`np.ones, np.zeros, np.eye, np.identity`).
Code de fonctions `SommeMatrices, ProduitMatrices,TransposeMatrice` qui implémentent ces opérations matricielles, fonction `TestCommutation` qui teste si deux matrices données commutent.

20.18 Opérations Numpy coefficients par coefficients (`+,-,/,*,**,`). Les fonctions réelles de Numpy s'appliquent coefficient par coefficient sur une matrice. Commande `np.vectorize`. Effet de $x * A$ et $x + A$ lorsque x est une variable numérique et A une matrice. Opérations matricielles `np.dot` et `np.transpose`.

Quelques questions de cours

1. Définir les notions de graphe (cas orienté ou non : au choix de l'interrogation), de sommets adjacents, isolés, de boucle, de graphe simple, d'ordre d'un graphe, les notions d'extrémités d'une arête et de degrés. Donner la définition ensembliste du graphe représenté ci-dessous (au choix de l'interrogation).
2. Énoncer et démontrer le lemme des poignées de mains (cas orienté ou non : au choix de l'interrogation). Est-il possible que, dans une assemblée de 16 personnes, une personne soit amie avec tout le monde et chacune des autres personnes soit amie avec exactement 4 personnes ?
3. Définir la notion de chaîne (ou de chemin : au choix de l'interrogation) et la notion de graphe connexe (dans le cas orienté ou non, en cohérence avec la définition précédemment demandée). Donner un exemple de graphe connexe, un exemple de graphe non connexe.
4. Définir la notion de matrice d'adjacence d'un graphe (orienté ou non, au choix de l'interrogation) et illustrer la définition sur l'exemple suivant (au choix de l'interrogation). Que dire de "la" matrice d'adjacence d'un graphe non orienté? Le démontrer.
5. Énoncer la proposition donnant l'interprétation des coefficients de M^k, où M est une matrice d'adjacence d'un graphe G orienté et où $k \in \mathbb{N}$. Démontrer ce résultat.
6. Énoncer le théorème (26) qui caractérise la connexité d'un graphe orienté à l'aide d'une matrice d'adjacence. Énoncer le lemme utilisé dans la preuve, et démontrer ce lemme.
7. Énoncer le théorème (26) qui caractérise la connexité d'un graphe orienté à l'aide d'une matrice d'adjacence. Énoncer le lemme utilisé dans la preuve, et démontrer ce théorème en admettant ce lemme.
8. Définir la notion de graphe complet. On note K_n le graphe complet d'ordre n dont les sommets sont les entiers de 1 à n. Donner la matrice d'adjacence M de K_n (pour la numérotation naturelle de ces sommets) et calculer M^p pour tout entier naturel p.
9. Écrire le code d'une fonction `ProduitMatrices` prenant en entrée deux matrices A et B (de type `np.ndarray`) telles que le produit AB soit correctement défini, et renvoyant en sortie la matrice AB.