Poirier Maxime
Karagoz Schahin

Projet :
Séquence Destructrice

Nous avons choisi d'implémenter le projet en python. Pour exécuter le programme, taper la commande « python projet.py » le programme attend ensuite les valeurs choisis de n et de p.

Pour l'implémentation nous avons choisi de faire un objet Graphe() qui contient un dictionnaire représentant le graphe sous forme de listes d'adjacence ainsi qu'une liste dont les indices sont les sommets des graphes et la valeur 1 correspond à un sommet colorié en rouge et 0 en bleu.
Cet objet possède les méthodes generation(n,p), qui génère un graphe quelconque, colorie(r), qui colore en fonction de r les sommets du graphe, ainsi que estDestructrice(seq) qui teste si la séquence seq est 2-destructrice ou non.

Référence de l'objet :

https://codes-sources.commentcamarche.net/source/47579-graphes-non-orientes?fbclid=IwAR2FrhJeu4ZoaVtYGH1V5YFBtH4gbPpQI3jfi_-ZmNXkGO8ZY3lyGrNm6y8

<u>Questions du sujets :</u>

1) Pour la figure (b), (v3,v2,v5,v6,v1,v4,v8,v7) est une séquence 2-destructrice.

3) Cette fonction parcours chaque sommet de la séquence et compte le nombre de voisins rouges du sommet dans le graphe qui ne sont pas déjà apparus précédemment dans la séquence. Si ce nombre est supérieur à 2 alors la séquence n'est pas 2-destructrice puisque il reste plus de 2 sommets voisins rouges à faire apparaître dans la séquence. Sinon on continue.
La complexité est de l'ordre de O(n*m)

4) Pour construire une séquence 2-destructrice, notre algorithme commence par construire une liste du nombre de voisins rouge de chaque sommets, puis tant que chaque sommet n'est pas pris dans la séquence, il prend le sommet ayant le minimum de voisins rouge (Si il n'y a pas de sommet ayant au plus 2 voisins rouge alors, une telle séquence n'existe pas et l'algorithme retourne une liste vide), puis si le sommet retenu est rouge, il décrémente le nombre de voisins rouges dans la liste des voisins de ce sommet (le nombre de voisins rouge du sommet retenu passe à +inf pour ne pas reprendre ce sommet).

7) Le degré attendu d'un graphe quelconque obtenu par le modèle G(n,p) est de l'ordre de p*(n-1).

Car chaque arêtes a une probabilité p d'appartenir au graphe et chaque sommets peut potentiellement avoir n-1 arêtes.

9)

	P = 0.1	P = 0.3	P = 0.5	P = 0.7
n=50	R = 65.62 %	R = 23.04 %	R = 14.45 %	R = 10.54 %
n=100	R = 32.91 %	R = 11.32 %	R = 7.22 %	R = 7.32 %