2018 - 2019

Algorithme et complexité : Rapport

Binômе :

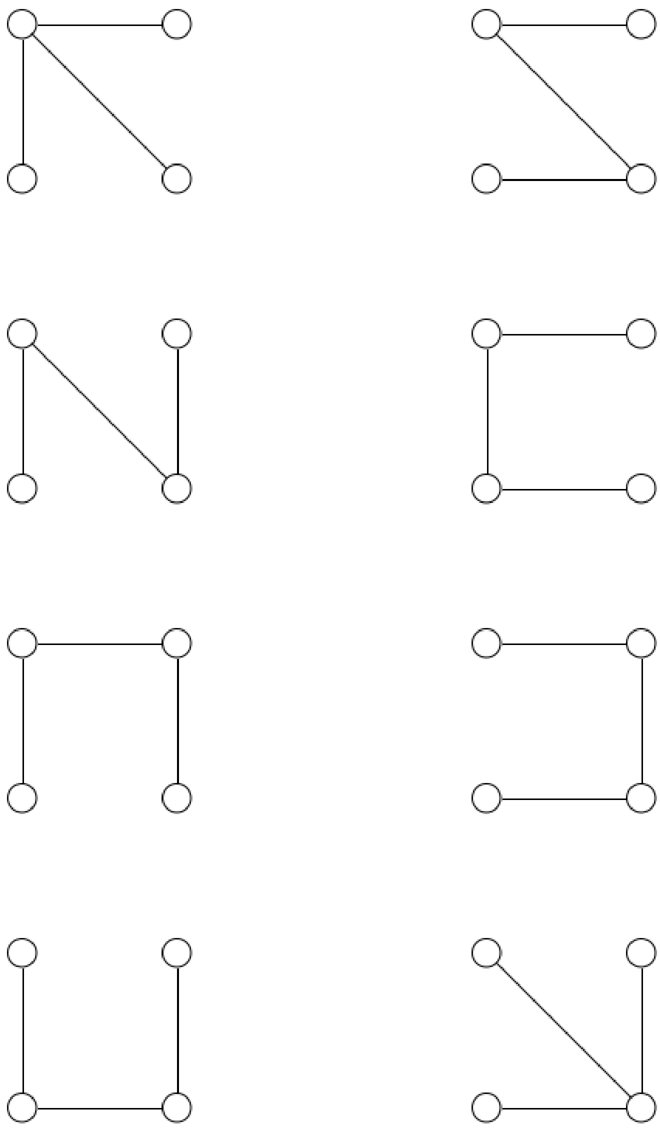
- *Bellanger Clément*
- *Fornito Marvin*

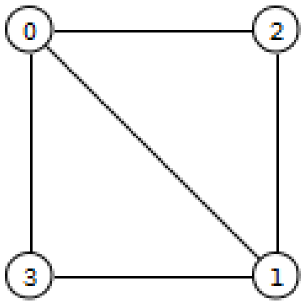
Lien Github :https://github.com/fornito2u/Algo-2019

Partie 1: Arbre Couvrant

Question 1 :

Voici les 8 arbres couvrants possibles pour l'exemple donné.





1.1) Algorithme de Kruskal

Question 2 :

L’algorithme de Kruskal est implémenté dans la classe “Kruskal” via la structure Union-Find implémenté dans la classe “UnionFind”.
Voici un exemple de résultat d'exécution : 0--3 0--1 1--2
Il y a 8 résultats possibles.

Question 3 :

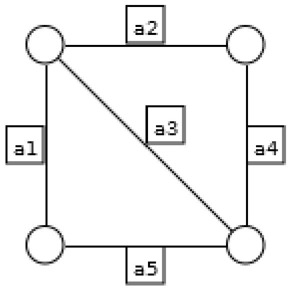
Comme nous pouvons le constater dans l’exemple ci-dessous, les 8 arbres couvrants n’ont pas tous la même probabilité d’apparaître (La colonne Arbre Q4 est à titre indicatif, elle n'apparaît pas dans le résultat de l'exécution) :

Lancement de l'algorithme Kruskal 1000000 de fois sur le graph donné

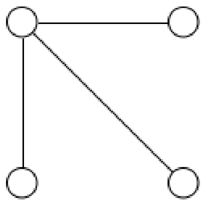
Numéro	Arbre	Nombre D'Occurrences	Probabilité	(Arbres Q4)
1	1--2 1--3 0--3	117162	0.117162	7
2	1--2 0--2 0--3	116149	0.116149	5
3	0--2 1--3 0--3	116301	0.116301	4
4	1--2 0--2 1--3	116476	0.116476	6
5	1--2 0--1 0--3	133547	0.133547	3
6	0--1 0--2 0--3	133347	0.133347	1
7	0--2 0--1 1--3	133890	0.13389	2
8	1--3 1--2 0--1	133128	0.133128	8

Question 4 :

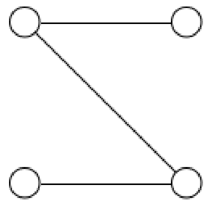
Tout d’abord on va commencer par numéroter toutes les arêtes possibles ainsi que les tous les arbres couvrants. On utilisera cette référence :



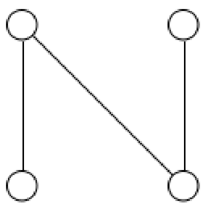
1



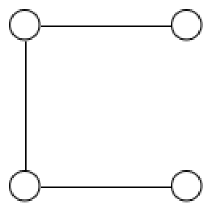
2



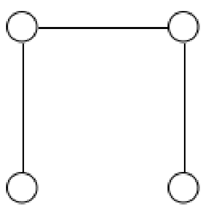
3



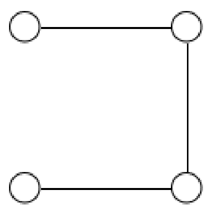
4



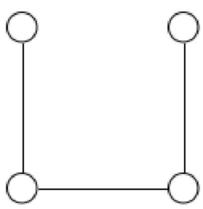
5



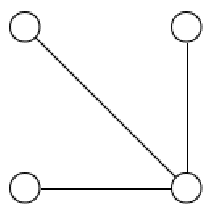
6



7



8



Dans l’algorithme de Kruskal, les arêtes sont choisies aléatoirement et ne sont enlevées que si elles forment un cycle. Il faut au minimum 3 arêtes pour faire un cycle.
On peut donc considérer que les 2 premières arêtes seront placées quoi qu’il arrive.
Or ces deux arêtes sont choisies avec la même probabilité.
On en déduit que chaque couple d’arêtes choisis au début par l’algorithme de Kruskal à la même probabilité d'apparaître.
Il y a 10 couples possibles.
Il suffit maintenant de noter la probabilité d’apparition de chaque arbre couvrant en fonction du couple de départ.
(Il y aura en fait 2 cas : soit il y a 3 possibilités, soit une arête forme un cycle et il ne reste que 2 possibilités pour les arbres couvrants)

Couple d'arêtes	Arbres possible	Probabilité
a1 a2	Arbres : 1,4,5	⅓ pour chaque arbre
a1 a3	Arbres : 1,3	½ pour chaque arbre
a1 a4	Arbres : 3,5,7	⅓ pour chaque arbre
a1 a5	Arbres : 4,7	½ pour chaque arbre
a2 a3	Arbres : 1,2	½ pour chaque arbre
a2 a4	Arbres : 5,6	½ pour chaque arbre
a2 a5	Arbres : 2,4,6	⅓ pour chaque arbre
a3 a4	Arbres : 3,8	½ pour chaque arbre
a3 a5	Arbres : 2,8	½ pour chaque arbre
a4 a5	Arbres : 6,7,8	⅓ pour chaque arbre

Comme on a listé tous les couples de départ possibles, on peut calculer la probabilité de chaque arbre d'apparaître (utilisation des probabilités totales).

Soit p(i) : La probabilité que l’arbre couvrant i apparaisse.

$p(1) = p(2) = p(3) = p(8) = 1/10*\frac{1}{2} + 1/10*\frac{1}{2} + 1/10*\frac{1}{3} = 8/60$ ou $2/15$
$p(4) = p(5) = p(6) = p(7) = 1/10*\frac{1}{2} + 1/10*\frac{1}{3} + 1/10*\frac{1}{3} = 7/60$

On constate donc que tous les arbres couvrants n’ont pas la même probabilité d'apparaître.

1.2) Algorithme d'Aldous-Broder

Question 5 :

On constate que les 8 arbres couvrants ont tous approximativement la même probabilité d'apparaître (⅛ soit 0.125) :

Lancement de l'algorithme Aldous-Broder 1000000 de fois sur le graph donné.

Numéro	Arbre	Nombre D'Occurences	Probabilité	(Arbres Q4)
1	1--3 0--3 1--2	125256	0.125256	7
2	1--2 1--3 0--2	124718	0.124718	6
3	0--1 1--2 0--3	125385	0.125385	3
4	0--3 1--3 0--2	124905	0.124905	4
5	0--1 0--2 0--3	124807	0.124807	1
6	1--2 0--2 0--3	124785	0.124785	5
7	0--2 0--1 1--3	124809	0.124809	2
8	1--2 0--1 1--3	125335	0.125335	8

1.3) Algorithme de Wilson

Question 6 :

1.4) Application ludique : Les Labyrinthes

Question 7 :

Question 8 :

Partie 2 : Mastermind

2.1) Question préalables

Question 9 :

Question 10 :

Question 11 :

Question 12 :

2.2) Programmation dynamique

Question 13 :

Question 14 :

Question 15 :

Question 16 :

2.3) Algorithme glouton

Question 17 :

Question 18 :

Question 19 :

Question 20 :

Partie 3 : Répartition du travail

L'ensemble des questions ont été réalisé en binôme. L'implémentation des algorithmes à était fait en parallèle par les deux membres du binôme en utilisant git comme gestionnaire de version.

Le rapport a également été rédigé par les deux membres du groupe, cette fois ci via l'utilisation d'un répertoire drive permettant l'écriture collaborative.