

# Leçon 925 – Graphes : représentations et algorithmes.

17 juillet 2019

## 1 Extraits du Rapport

### Rapport de jury 2018

*Cette leçon offre une grande liberté de choix au candidat, qui peut décider de présenter des algorithmes sur des problèmes variés : connexité, diamètre, arbre couvrant, flot maximal, plus court chemin, cycle eulérien, etc. mais aussi des problèmes plus difficiles, comme la couverture de sommets ou la recherche d'un cycle hamiltonien, pour lesquels il pourra proposer des algorithmes d'approximation ou des heuristiques usuelles. Une preuve de correction des algorithmes proposés sera évidemment appréciée. Il est attendu que diverses représentations des graphes soient présentées et comparées, en particulier en termes de complexité.*

## 2 Cœur de la leçon

- Représentations des graphes, complexités des opérations élémentaires.
- Parcours en largeur et en profondeur.
- Plus courts chemins, Dijkstra.

## 3 À savoir

- Plus courts chemins, Floyd Warshall, Bellman Ford.
- Arbres couvrants, Prim, Kruskal.
- Fermeture transitive d'une relation. (notamment par adaptation de Floyd Warshall)

## 4 Ouvertures possibles

- Algorithmes de composantes connexes (Kozaraju, Tarjan)
- Cycle eulérien, hamiltoniens.
- Problèmes de flots.
- Lien avec la complexité et la  $NP$ -complétude.

## 5 Conseils au candidat

- Faire des dessins.
- Avoir écrit des algos dans le plan peut permettre d'éviter des les réécrire pendant un développement.
- Ne pas essayer de faire un plan trop original. Cette leçon est une liste d'algorithmes ! Il faut cependant essayer de justifier l'ordre de présentation, et mettre du liant.
- Donner des applications aux algorithmes (approximations de problèmes  $NP$ , etc ...)

- Ne pas faire une partie "problèmes NP", ou plus généralement sur la théorie des graphes.
- Ne pas se restreindre aux graphes orientés.
- Pour les graphes pondérés, ne pas prendre des poids réels, uniquement des entiers.

## 6 Questions classiques

- Exemple d'application de la détection de cycle, de flots, de ... ?
- Pourquoi les graphes sont-ils si importants ?
- Dérouler tel algorithme sur tel exemple.
- Qu'est-ce qui peut influencer le choix entre les différents algorithmes de plus courts chemins.
- Est-ce que le choix de la représentation influe fortement sur la complexité en espace des algorithmes ?
- Donner un exemple d'application des tris topologiques ?

## 7 Références

- [Cor] Algorithmique - CORMEN - à la BU/LSV  
*La bible de l'algorithmique, avec toutes les bases. Attention, les calculs avec des probas sont parfois faux.*
- [Bea] Éléments d'algorithmique - D. BEAUQUIER, J. BERSTEL, Ph. CHRÉTIENNE - à la BU/LSV  
*Bonne référence pour l'algo, pleins de dessins et de preuves. Un peu vieillissant et devenu rare.*

## 8 Dev

- ++ Correction totale et/ou complexité de DIJKSTRA - ([Cor], [Bea], p. ?) - 925,926,927  
*Long de faire correction et complexité, doit-être adapté selon la leçon.*
- ++ Correction des algorithmes de Prim et Kruskal - (Cor, p. ??) - 925,927  
*La preuve du Cormen est générale pour ce type d'algorithme.*
- 2SAT est NL-Complet (ou algo en temps linéaire selon les leçons) - ([Car], [Pap], [Cor] (pour les algo de graphe)) - 915,916,925 *Bien insister sur les algorithmes de graphes utilisés (Kosaraju par exemple), et bien justifier que le temps reste linéaire en la taille de la formule.*