

UNIVERSITE MOHAMED KHIDER – BISKRA  
FACULTE DES SCIENCES EXACTES, DES SCIENCES DE LA NATURE ET DE LA VIE  
DEPARTEMENT D'INFORMATIQUE

# Complexité et Optimisation

---

**Support de cours et TD**  
Master d'informatique

Réalisé par : Dr. S. SLATNIA

**Version améliorée**

05/01/2016

# OBJECTIFS DE COURS

L'objectif de ce cours est de trouver un bon algorithme pour un problème donné. La résolution de ce problème revient à comparer les performances de deux algorithmes effectuant les mêmes tâches, deux paramètres essentiels parmi les ressources coûteuses sont à prendre en compte : le temps d'exécution (Complexité temporelle), et l'espace mémoire requis (Complexité spatiale) qui permet de représenter la quantité de mémoire nécessaire pour l'exécution d'un programme.

La complexité temporelle, pour rendre l'étude du coût temporel d'un algorithme indépendante de tout aspect physique, il faut prendre pour unité de mesure une opération élémentaire pertinente pour le problème étudié. Calculer *le coût (la complexité)* d'un algorithme, est de déterminer *le nombre d'opérations élémentaires effectuées par cet algorithme*. Cette comparaison doit être indépendante de l'ordinateur utilisé, et sur un même ordinateur du langage de programmation utilisé pour rendre la comparaison indépendante de tout aspect technique.

Parmi les problèmes qui permettent d'engendrer des valeurs de complexité très grande, c'est les problèmes qui nécessitent d'explorer des espaces de recherche très grands, on utilise la récursivité. Les problèmes de tri et de recherche permettent de présenter des meilleurs exemples de ce type de complexité. La récursivité est un des concepts de programmation les plus importants. Le principe de l'approche récursive est de ramener le problème à résoudre à un sous-problème correspondant à une instance «réduite» du problème lui-même.

Autrement dit, La recherche d'algorithme ayant une complexité plus petite que les meilleurs algorithmes connus est un thème de recherche important dans toutes les branches de l'informatique. La réalisation de cet objectif nécessite des connaissances d'approfondir qui concerne le domaine de l'analyse d'algorithmique et les techniques d'optimisation de la complexité présentés par l'organisation suivante :

## OBJECTIFS DE COURS

---

Le premier chapitre est consacré à l'étude de la théorie de la complexité et mesure de performance. Le chapitre est divisé en deux parties, la première partie est consacrée à la théorie de la complexité, critères d'évaluation de performance de l'algorithme, complexité d'un algorithme, complexité d'un problème, type de complexité, notion d'optimalité. Et la deuxième partie est consacrée à l'étude de grandeur des fonctions, la complexité asymptotique et calcul de complexité des algorithmes itératifs et récursifs, cette partie permet de présenter les concepts fondamentaux de la complexité des algorithmes, optimisation d'exécution de la récursivité, c'est quoi les notions de Landau ? Et les classes de complexité et notations asymptotique, analyser le coût des algorithmes récursifs en utilisant le Master théorème et les équations de récurrences.

Le deuxième chapitre est pour définir l'étude des classes de complexité des problèmes. Le chapitre permet de présenter la complexité d'un problème, trois méthodes pour trouver une borne inférieure, le chapitre permet de définir les classes de complexité d'un problème : classes de complexité en temps, Classe P, NP et EXPTIME et classes de complexité en espace. Ce chapitre permet ainsi de présenter les NP-complétude et les notions de base de la réduction d'un problème Q à un problème  $\Pi$ .

Le troisième chapitre pour objectif de présenter la stratégie diviser pour régner, optimisation de la complexité, l'élément majoritaire et la dichotomie. Le chapitre permet de présenter encore l'optimisation et algorithmes avancés, une comparaison de complexité des algorithmes de tri et de recherche.

Le chapitre quatre est consacré à la présentation de l'optimisation combinatoire, méthodes exactes, définition d'optimisation, définition de problème d'optimisation, programmation dynamique, Branch and bound, recherche arborescente, l'exploration sans information. Les approches récursives se trouvent dans les parcours d'arbres, les recherches en largeur et en profondeur d'abord. Et en particulier, ce chapitre permet de présenter l'exploration des graphes et les algorithmes de recherche en profondeur, application du backtracking dans les arbres de jeux.

Nous clôturons la présentation de ce cours par le cinquième chapitre qui permet de présenter la notion d'optimisation combinatoire, quelques définitions de base et les méthodes heuristiques de résolution d'un problème d'optimisation. Nous focalisons notre étude sur les algorithmes de recherche locale, Glouton,  $A^*$ , Hill Climbing,