

Algorithmique et complexité

2022-2023

Préambule

- Téléphone toléré uniquement à usage pédagogique
- Présence intellectuelle indispensable
- Être actif, proactif et impliqué
- La prise de notes est indispensable (examen = A4 R/V)
 - <http://labouseur.com/commondocs/Against-Laptops-and-Phones-in-Class.pdf>
- Il y a un cours, des transparents, codes, lectures, ...

Internet c'est bien mais bon ...,

 - Utilisez le cours comme point de départ
 - La démarche pédagogique (énoncé, ...) se construit sur cette trame de cours (ça peut aider à comprendre les énoncés ..)
 - Il n'y a pas de moteur de recherche dans mes transparents → à vous d'indexer le cours ☺

Koikizendize (20-21)

➤ Points forts

- Le fonctionnement préparation TD et CM est très intéressant. De plus les explications sont claires et le cours bien adapté a ce contexte particulier lié au covid.
- les séances des Tds
- bonnes explications du prof à l'oral
- avoir à travailler sur des exemples concrets
- La partie préparation de CM et de TD est une très bonne idée.
- Cours très intéressant qui invite à sortir un peu des limites d'un cours classique et de pousser la réflexion. La pédagogie développée par Mr Chevrier est intéressante.
- Bonne méthode pédagogique

Koikizendizenkor (20-21)

➤ Points améliorations

- La difficulté entre les TD est assez hétérogène. Le TD 3 est bien plus dur que les autres (*TD 3 = glouton !!!*).
- De plus certains points ont déjà été vus l'année dernière comme les algorithmes gloutons, la force brute . Y passé moins de temps en les réunissant par exemple pourrait permettre de couvrir plus de choses
- oui mais selon les promos, la maîtrise des concepts est variable d'où une revue. Je vais essayer d'aller plus vite aux fondamentaux et leur mise en œuvre
- ajouter des Tps
- Ce sera fait (en présentiel et en Travail perso)

.....

- questions des td et du sujet de partiel parfois peu claires
- visuel mériterait d'être plus remplis pour avoir une trace de certains points importants
- Rendre les sujets de préparations et de TD plus clairs.
- Pas spécialement de point à améliorer, je pense que le cours est bien comme il est
- Cours facile de s'y perdre à cause de la gestion des polices et tailles des titres principaux

Retour 21-22

- Cf document
- Conséquence:
 - Vous devez rattraper vos *absences* (je ne vous cours plus après)
 - Consignes données en CM
 - Affectation des problèmes en TD
 - Tout ce qui écrit au tableau !
 - Etc.
 - Vous devez anticiper vos problèmes
 - Ce n'est pas parce que c'est vu en 1A que vous maitrisez 😊

Présentation

- 10h CM
- 10h TD
- Contrôle des connaissances
 - Examen écrit 2
 - Compte rendu de TD/ TP/travail perso 1 (non rattrapable/respecter les délais et s'assurer de la bonne réception)
- Enseignant Vincent Chevrier

Explication de texte

- non rattrapable
 - Si mauvaise note, elle ne pourra plus être changée
- respecter les délais et s'assurer de la bonne réception
 - Pour mettre une note il faut recevoir un travail dans les délais
 - ➔ si je n'ai rien, c'est zéro
 - ➔ « il fallait me dire que vous n'aviez rien reçu » ➔ pas recevable (dépôt sur arche)
 - ➔ On doit savoir quand on est en retard, le travail s'organise (on peut **prévenir** et échanger sur un possible retard)
Prévenir donc avant la deadline

Objectifs

- **Comprendre** ce que signifie complexité algorithmique
- Savoir reconnaître les grandes classes de complexité
- Etudier des algorithmes efficaces face à des problèmes prototypiques (tri, recherche, etc..)
- Connaître et savoir appliquer les stratégies de construction d'algorithmes

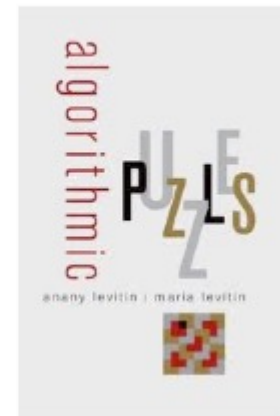
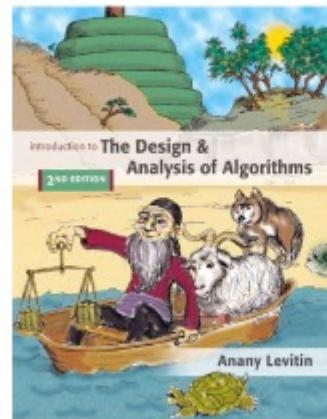
Plan

- Introduction
 - Notion de complexité
 - Éléments de réflexion
- Construction d'algorithmes
 - Force brute
 - Approche gloutonne
 - Diviser pour régner
 - Programmation dynamique
 - Retour arrière
 - *Approche heuristique en Intelligence artificielle*

Bibliographie

- C. Froidevaux, M.-C. Gaudel et M. Soria,
Types de données et algorithmes,
Mc Graw Hill, 1990
- T.H. Cormen, C.E. Leiserson et R.L. Rivest,
Introduction to Algorithms,
MIT Press, Mc Graw Hill, 1989
- Voir aussi

Anany Levitin. **Introduction to the Design and Analysis of Algorithms.**
Edition, Addison-Wesley. 2003.



Evaluation partie contrôle continu

➤ Objectif :

- Apprendre à *réfléchir* autour des notions du cours
- Maîtriser les concepts et les mettre en oeuvre sur des cas non traités ou traités partiellement (**donc codage**)

Mise en garde

- Toute tentative de plagiat sera sanctionnée par un zéro
 - Reprise d'exemples, de texte, de données à partir d'internet
 - Code pas cohérent avec le niveau observé en TD
 -
- **Mais** on peut citer ses sources, expliquer sur quelles parties du travail on s'est fait aider,

Evaluation contrôle continu

- Travail "spontané" sur la préparation des cours et TD (pas de note)
- Travail individuel
 - **imposé** autour des concepts du cours et TD
 - Enoncé imposé par l'enseignant
 - Tout retard de rendu sera sanctionné (vous êtes **responsables**)
 - Rendu sur arche (plusieurs dépôts possibles)
- Travail en groupe (4)

Précisions : travail personnel imposé

➤ Comment

- Fournir un compte rendu (voir plus loin)
- Aller au delà de ce qui est vu
 - Extensions, généralisations d'un cas traité
 - Exercice(s) non traité(s) (complètement) en TD
- Livrables
 - CR (5 pages maxi hors algo)
 - Un algorithme + un générateur de données (cf TP)

Compte rendu

(T0)

- Enoncé imposé

(T0 + 2jours) (suggestion)

- Rappel du problème

(T0 + 1semaine) (suggestion)

- Analyse et représentation adoptée, choix, justification, ..
- Algorithme

(T0 + 2semaines) [OBLIGATOIRE]

- Code
- Test
- Discussion et extensions

- Le document de départ doit être enrichi, cohérent et pas réécrit à chaque étape (sauf justification)

TP: évaluation empirique de complexités algorithmiques

- Par groupe de 4/5, en lien avec travail autour des TD
- Idée directrice :
 - construire un code le plus générique possible qui permet d'estimer expérimentalement la complexité d'un algorithme,
 - l'appliquer dans votre groupe sur les TD et comparer les complexités/les approches d'évaluation
- Exemple :
 - Pièce de monnaie (traité au tableau)

Compte rendu (aperçu)

- Rappel du problème
 - Les N reines ...
- Analyse et représentation adoptée, choix, justification, ..
 - Approche force brute pour $N = 5$ avec implantation directe (pas de représentation de l'espace d'état)
 - On itère pour chacune des reines sur ses positions possibles
 - On teste si on a une solution ou pas

Compte rendu (aperçu)

➤ Algorithme

- Pour reine1 de 1 à 5
 - Pour reine2 de 1 à 5
 - Pour reine3 de 1 à 5
 - » Pour reine4 de 1 à 5
 - » Pour reine5 de 1 à 5
 - » Si NON enprise (reine1, reine2, reine3, reine4, reine5) alors
 - » Ecriresolution()
- Enprise retourne vrai si une reine (au moins) est enprise avec une autre

Compte rendu (aperçu)

➤ Code

- ... on explique comment on programme cela
- en allant à l'essentiel (le lecteur est supposé avoir le même niveau que vous en programmation)

➤ Test

- Ce qui marche, ce qui ne marche pas, complexité expérimentale, ...

➤ Discussion et extension

- Ce qui est bien fait : on trouve les solutions
- Améliorations : passer à N quelconque (comment), s'arrêter dès la première solution trouvée, récupérer toutes les solutions, etc

Compte rendu (aperçu)

- Pour aller vers le TP
 - Envisager de générer des données pour tester ses hypothèses
 - Illustration sur le pb des pièces de monnaie