

Bibliographie informatique agreg' maths option info 2020

Par **Lilian Besson**. Toutes remarques sont les bienvenues par courriel.

On liste ici les livres très utilisés comme références pour préparer les leçons et les développements, ainsi que l'épreuve de modélisation de l'option informatique D.

Plus d'informations sur la prépa agreg' maths option info 2020 : sur le wiki du DPT info de l'ENS de Rennes.

Plus de détail sur chaque livre sont disponibles via le `bibliopac.ens-rennes.fr`).

Aussi en PDF `info.pdf`, en maths (`maths.pdf`) et combinés (en PDF) !

Notez qu'une bibliographie plus concise est disponible ici : smallbib. Et une bibliographie minimaliste est disponible ici : smallsmallbib.

Si vous voulez convertir cette liste en une bibliographie BibTeX, je serai intéressé !

Quelques pages web sur l'option D par des anciens élèves récents

- Julie Parreaux (2019)
- Pierre Le Barbenchon (2019), regardez sa bibliographie à la fin de son (gros) document et ses livres préférés
- Clarence Kineider (2019)
- Joshua Peigner (2018)
- Aude Le Gluher (2017)
- Théo Pierron (2014) (un bon doc)
- Vous êtes invités à faire pareil (c'est très formateur !)

En Informatique

Voici une liste assez complète de livres utiles pour préparer les deux épreuves d'informatique de l'agrégation de mathématiques (option D) : oral de leçons d'informatique (21 leçons en 2020) et oral de modélisation d'option D.

J'indique entre accolades les leçons couvertes par chaque livres (ce sont des indications rapides, des livres peuvent très bien être utiles pour d'autres leçons).

TODO liste inversée

Algorithmique

- **901** : Structures de données : exemples et applications. : [[TODO]],
- **903** : Exemples d'algorithmes de tri. Complexité. : [[TODO]],
- **907** : Algorithmique du texte : exemples et applications. : [[TODO]],
- **921** : Algorithmes de recherche et structures de données associées. : [[TODO]],
- **925** : Graphes : représentations et algorithmes. : [[TODO]],
- **926** : Analyse des algorithmes : complexité. Exemples. : [[TODO]],
- **927** : Exemples de preuve d'algorithme : correction, terminaison. : [[TODO]],
- **931** : Schémas algorithmiques. Exemples et applications. : [[TODO]],

Langages Formels

- **909** : Langages rationnels. Exemples et applications. : [[TODO]],
- **923** : Analyses lexicale et syntaxique : applications. : [[TODO]],

Logique

- **916** : Formules du calcul propositionnel : représentation, formes normales, satisfiabilité. Applications. : [[TODO]],
- **918** : Systèmes formels de preuve en logique du premier ordre : exemples. : [[TODO]],
- **924** : Théories et modèles en logique du premier ordre. Exemples. : [[TODO]],

Calculabilité et complexité

- **912** : Fonctions récursives primitives et non primitives. Exemples. : [[TODO]],
- **913** : Machines de Turing. Applications. : [[TODO]],
- **914** : Décidabilité et indécidabilité. Exemples. : [[TODO]],
- **915** : Classes de complexité : exemples. : [[TODO]],
- **928** : Problèmes NP-complets : exemples de réductions : [[TODO]],
- **929** : Lambda-calcul pur comme modèle de calcul. Exemples : [[TODO]],

Autres domaines

- **932** : Fondements des bases de données relationnelles. : [[TODO]],
- **930** : Sémantique des langages de programmation. Exemples : [[TODO]],

Livres spécifiques pour l'oral de l'agrégation

[131dev] “**131 développements**” (par **Le Barbenchon, Lesevre, Pierron, Montagnon**)

Une bible pour les développements de l'agrégation, y compris en informatique ! Contient 131 développements de niveaux variés, couvrant toutes les leçons du programme de 2019. Mon livre préféré parmi ceux pour l'oral !

Livres génériques pour l'option info

[Dehornoy] “**Mathématiques de l'informatique : cours et exercices corrigés**”

Un excellent bouquin, qui présente rapidement (presque) tous les éléments du programme, avec cours, exemples, démonstrations et exercices ! Contient plein de développements parmi les plus utiles ou les plus classiques. Mon livre préféré (parmi ceux pour l'option info) !

[Stern] “**Fondements mathématiques de l'informatique**”

Un bon bouquin, mais qui a mal vieilli. Encore de bonnes démos, notamment des réductions pour les problèmes NP-complets.

[Belghiti] “**Les clefs pour l'info**”

Un recueil d'annales corrigées de l'épreuve d'informatique théorique du concours des ENS. Abordant des domaines aussi variés que la logique propositionnelle, la combinatoire des mots, l'algorithmique, la réécriture ou la théorie des graphes, ces exercices permettent un survol complet d'un domaine peu abordé dans la littérature accessible à Bac+2. Outre les corrections détaillées et illustrées de nombreux schémas, les auteurs ont fourni des commentaires et une bibliographie raisonnée.

C'est une référence qui contient plein d'exercices corrigés, chacun pouvant être utilisé comme un développement. Le niveau est très haut, attention à ne pas viser trop difficile.

[Jaume] “**Éléments de mathématiques discrètes : cours, exercices résolus, implémentations avec les langages Python et OCaml**”

Un bon livre de mathématiques discrètes pour l'informatique, qui parle de beaucoup de notions importantes pour l'agrégation : ensembles, relations, fonctions, structures ordonnées, définitions inductives, langages, automates, et logique. Chaque notion est illustrée sur de nombreux exemples (plus de 150 exemples sont fournis), fait l'objet de nombreux exercices corrigés (144 exercices sont fournis) et les étapes de raisonnement de chaque preuve sont explicitées dans les moindres détails. Enfin, une implantation avec les langages Python et OCaml est fournie pour la plupart des concepts effectifs introduits. Il s'agit de montrer, au travers de 250 programmes, comment exprimer un calcul à l'aide d'un langage de programmation et de permettre une expérimentation de ces calculs afin d'en obtenir une compréhension profonde.

Cette référence peut être très utile pour l'oral de modélisation, car elle montre beaucoup d'exemples de codes, propres, concis et récents, en OCaml et Python.

Informatique niveau prépa (option info et info pour tous)

[Wack] “**Informatique pour tous en classes préparatoires aux grandes écoles : manuel d'algorithmique et programmation structurée avec Python**”

Un excellent livre qui couvre tout le programme d'**Informatique pour tous** en prépa. Très utile pour les rappels de cours sur l'architecture des ordinateurs (modèle de von Neumann), la représentation des nombres entiers/rationnels/flottants en machine, ainsi que pour les bases de données et les bases du calcul scientifique (méthode d'Euler, pivot de Gauss etc).

[Albert, Gastin] “**Cours et exercices d'informatique : classes préparatoires**”

Un excellent livre qui couvre tout le programme d'**option Informatique** en prépa MPSI/MP. Avec beaucoup de programmes `Cam1`, mais peu de preuves. De bons rappels de cours sur les bases, souvent illustrés par un peu de code. Peut vraiment aider pour l'épreuve de modélisation !

[Jaume] “**Éléments de mathématiques discrètes : cours, exercices résolus, implémentations avec les langages Python et OCaml**”

Un bon livre sur le domaine des mathématiques discrètes, véritable frontière entre mathématiques et informatique. Gros avantage de donner des implémentations en Python et en OCaml, c'est une source rare pour travailler ces deux langages conjointement pour l'épreuve de modélisation.

Algorithmique 901,903,907,921,925,926,927,931

Couvrent les leçons suivantes :

- **901** : Structures de données : exemples et applications.
- **903** : Exemples d'algorithmes de tri. Complexité.
- **907** : Algorithmique du texte : exemples et applications.
- **921** : Algorithmes de recherche et structures de données associées.
- **925** : Graphes : représentations et algorithmes.
- **926** : Analyse des algorithmes : complexité. Exemples.
- **927** : Exemples de preuve d'algorithme : correction, terminaison.
- **931** : Schémas algorithmiques. Exemples et applications.

[BBC] “Éléments d’Algorithmique” 901,903,907,921,925,926,927,931

Une excellente référence, surtout sur les structures de données. **Attention** il devient rare. Contient presque tout, avec plein de dessins et plein de preuves. Il n’est plus édité et le jury autorise que les malles contiennent des versions reliées d’un poly imprimé (il est diffusé librement par ses auteurs).

[Cormen] “Algorithmique : cours avec 957 exercices et 158 problèmes” 901,903,907,921,925,926,927,931

La **bible de l’algorithmicien**, toujours précis et rigoureux pour ses preuves. Il convient de rester vigilant, quelques typos ou erreurs restent présentes, même dans la dernière édition. Certaines peuvent inspirer des développements, et certains algorithmes (hachages, arithmétique, ?) peuvent être présentés directement en développement de maths.

Il s’appelle enfin autrement que “introduction à l’algorithmique” (en 4ème édition), parce que le plus gros bouquin de la liste n’avait quand même plus de raison de s’appeler “introduction”...

[Aho, Hopcroft, Ullman] “Structures de données et algorithmes” 901,903,907,921,925,926,927,931

Une excellente référence (plus souvent disponible en anglais). **Pas disponible** à la BU de l’ENS de Rennes, mais dans la malle d’agrég. (peut-être à la BU de Beaulieu ?) De très bons rappels sur “tout”, en particulier les questions de dictionnaires, graphes (orientés ou non), et les tris. Une preuve presque claire de la borne inférieure du nombre de comparaisons pour un algorithme de tri par comparaisons.

[Baynat] “Exercices et problèmes d’algorithmique (146 énoncés avec solutions détaillées)” 901,903,907,921,925,926,927,931

Un bon bouquin, qui souvent vient avec une rédaction “type développement” et un découpage par lemmes. Notamment les chapitres 5 sur les bases des graphes, 6 sur les parcours et 7 sur les graphes valués.

[Monicault] “L’épreuve écrite d’algorithmique X, ENS, ESPCI : cours abrégé annales corrigées” 901,903,907,921,925,926,927,931

Un petit livre récent qui donne un bon condensé du programme d’informatique de prépa (ce qui couvre la moitié du programme d’agrég), et plein d’exercices et de problèmes corrigés. Peut donner des idées de développements originaux.

[Bajard] “Exercices d’algorithmique : oraux d’ENS” 901,903,907,921,925,926,927,931

Un livre un peu vieillissant (1997) qui peut donner des idées de développements difficiles pour les leçons d’algorithmiques. Je ne le connais pas bien.

[Boissonnat, Yvinec] “Géométrie algébrique”

Trop complet... Utile pour les questions de triangulations et les diagrammes de Voronoï. Présente plein de méthodes de calculs de l’enveloppe convexe (autre que Graham et Jarvis).

Algorithmes et implémentations

Utile pour s’entraîner à l’oral de modélisation.

[Durr] “Programmation efficace : les 128 algorithmes qu’il faut avoir compris et codés en Python au cours de sa vie”

Un petit livre qui s’attaque à définir plein de problèmes algorithmiques et présenter leurs solutions, en expliquant l’algorithme et en donnant à chaque fois un code complet en Python. Site web compagnon : <https://tryalgo.org/>. On y trouve les algorithmes classiques de géométrie ou de recherche de plus court chemin mais également des

sujets plus atypiques tels que les arbres de Fenwick ou les liens dansants de Knuth. C'est une très bonne référence pour travailler la programmation en Python et l'épreuve de modélisation.

[Conchon] “Apprendre à programmer avec OCaml - Algorithmes et structures de données”

Un livre d'algorithme utilisant OCaml, c'est une très bonne référence pour se réapproprier le langage OCaml et travailler les leçons d'algorithmique, ainsi que la modélisation.

Algorithmique du texte 907

Couvre la leçon suivante **907** (Algorithmique du texte : exemples et applications.).

[Crochemore] “Algorithmique du texte” 907

Présente tout ce qu'il faut savoir sur l'algorithmique du texte au niveau agrég, et bien plus. Présente notamment les algorithmes KM, KMP, les automates de Simon, mais aussi Boyer-Moore (qui est l'algorithme effectivement utilisé dans GNU grep), ainsi que le codage de Huffman, parmi d'autres choses.

[Crochemore, Rytter] “Text algorithms” 907

Version en anglais. Semble distribué en ligne légalement. Présente KM, KMP, les automates de Simon, mais aussi Boyer-Moore (qui est l'algorithme effectivement utilisé dans GNU grep), ainsi que le codage de Huffman, parmi d'autres choses.

Graphes 925

Couvrent la leçon **925** (Graphes : représentations et algorithmes.) et peuvent être utiles pour les leçons **901** (Structures de données : exemples et applications.) et **921** (Algorithmes de recherche et structures de données associées.), ainsi que **926** (Analyse des algorithmes : complexité. Exemples.) et **927** (Exemples de preuve d'algorithme : correction, terminaison.).

[Gondran, Minoux] “Graphes et algorithmes” 901,921,925,926,927

Un peu vieux, mais reste très complet. Présente les questions de connexité (et calcul des composantes connexes), de problème du plus court chemin (Moore-Dijkstra, Dijkstra, Bellman, Ford, Floyd, Dantzig etc), un bon chapitre sur les matroïdes, et un autre sur les arbres et arborescences (Kruskal, Prim).

[Fournier] “Théorie des graphes et applications : avec exercices et problèmes” 901,921,925,926,927

Un très bon bouquin, orienté applications. Il existe aussi en deux volumes (dans la malle d'agrég). Le tome 1 en particulier est très clair et complet avec de bons rappels sur les définitions, les questions de représentations des graphes, mais aussi la recherche de chemins optimaux, de d'arbres couvrants. Le tome 2 présente notamment le problème de voyageur de commerce.

Les livres suivants sont plus récents, je ne le connais pas encore.

[Bachir] “Graphes, ordres programmation linéaire : cours et exercices” 901,921,925,926,927

Un livre très récent (2020) sur les graphes. La première partie devrait être une bonne référence pour la leçon sur les graphes, et les exercices les plus difficiles peuvent donner des développements originaux.

[BoriesLonguet] “Graphes et combinatoire : cours avec 210 exercices corrigés” 901,921,925,926,927

Un livre récent (2015) sur les graphes. La première partie devrait être une bonne référence pour la leçon sur les graphes, et les exercices les plus difficiles peuvent donner des développements originaux.

[Larramendy] “Introduction à la théorie des graphes : cours et exercices corrigés” 901,921,925,926,927

Un livre récent (2015) sur les graphes. La première partie devrait être une bonne référence pour la leçon sur les graphes, et les exercices les plus difficiles peuvent donner des développements originaux. Peut aussi être utile pour des textes de modélisation qui parleraient de théorie des graphes (coloriages, classifications etc).

[Bretto] “Éléments de théorie des graphes” 901,921,925,926,927

Un autre livre récent (2012) sur les graphes. La première partie devrait être une bonne référence pour la leçon sur les graphes, et les exercices les plus difficiles peuvent donner des développements originaux.

Langages formels et automates 909

Concerne le **leçons 909** (Langages rationnels. Exemples et applications.) et les leçons de calculabilité (certains problèmes sur des classes de langages étant indécidables). Peut aussi contenir du matériel pour les leçons d'algorithmique, notamment **931** (Schémas algorithmiques. Exemples et applications.) avec l'algorithme de Cocke-Yasami-Kounger (CYK).

Il y a quelques années, il y avait plus de leçons sur les automates, avec notamment une leçon entière sur les automates dans laquelle on devait parler d'automate à pile, une notion qui est sortie du programme.

[CartonPerrin] “Langages formels, Calculabilité et Complexité : cours et exercices corrigés” 909,913,914,915,927

Une excellente référence. Beaucoup de développements, en langages formels bien sûr, mais aussi ailleurs (notamment problèmes NP et réductions). Mon autre livre préféré (parmi ceux pour l'option info) !

[Hopcroft, Ullman] “Introduction to automata theory, languages, and computation” 909

Une excellente référence (en anglais, mais dans une version internationale assez facile à comprendre). Beaucoup de rappels, d'exemples et de bons dessins à réutiliser pour les plans. De bonnes idées de développements sur les automates (Chap 2), les langages rationnels (Chap 3 et 4), algébriques (Chap 5 et 7), mais aussi de l'indécidabilité (Chap 9) des problèmes NP (Chap 10, dont `NodeCover`, `IndependentSets`, et `HamPath`) et une introduction à la classe Co-NP.

[Sakarovitch] “Éléments de théorie des automates” 909

Une bonne référence, même si son style austère rebute un peu. Très long. Très complet sur plein de choses hors programmes (youpi), mais aussi plein de choses sur ce qui est au programme des leçons d'info. Plein de développements sur tout ce qui concerne les automates, et un peu plus (PCP Th8.2 p42, des problèmes décidables sur les langages rationnels Prop1.11 p77, etc).

Jean-Michel Autebert : langages formels, et calculabilité

Des bouquins qui commencent à vieillir, mais restent de solides références.

[Autebert, Langages et Automates] “Théorie des langages et des automates” 909

[Autebert, Langages algébriques] “Langages algébriques” 909

Pas disponible à la BU de l'ENS de Rennes, mais dans la malle d'agrég. (peut-être à la BU de Beaulieu ?)

[Autebert, Langages algébriques] “Calculabilité et décidabilité : une introduction” 913,914,915

Décidabilité et calculabilité 912,913,914,915,928

Concernent les leçons suivantes :

- **912** : Fonctions récursives primitives et non primitives. Exemples.
- **913** : Machines de Turing. Applications.
- **914** : Décidabilité et indécidabilité. Exemples.
- **915** : Classes de complexité : exemples.
- **928** : Problèmes NP-complets : exemples de réductions.

Ces questions sont aussi abordées dans des livres cités ailleurs ([Dehornoy], [CartonPerrin], [Sakarovitch] etc). Les livres d'algorithmes ([Cormen] par exemple) contiennent souvent une partie entière sur des problèmes NP-complets et des algorithmes d'approximation, dont il convient de parler pour la leçon **928** (mais elle traite principalement de réductions et pas d'algo d'approximation, attention!).

[Wolper] “Introduction à la calculabilité” 912,913,914,915,928

Une excellente référence pour les leçons de calculabilité (des preuves bien rédigées, mais des exercices sans correction) et plein développements possibles : inclusion stricte $\mathcal{R} \subsetneq \mathcal{RE}$, problèmes indécidables sur les grammaires (Ch7.5 p207), etc. Prenez la dernière édition de 2006.

[Perifel] “Complexité algorithmique”

Un très bon livre sur les notions de calculabilité et complexité, des machines de Turing aux problèmes NP-complets, en passant par la hiérarchie polynomiale. Très bien rédigé dans un style clair et précis, avec de bons exemples et illustrations et plein d'exercices corrigés.

[DehornoyCalc] “Complexité et décidabilité” 912,913,914,915,928

Un autre bon bouquin sur la complexité et calculabilité. Contient des développements parmi les plus utiles ou les plus classiques.

[RidouxLesventes] “Calculateurs, calculs, calculabilité” 912,913,914,915,928

Un autre bon livre sur la calculabilité, qui a l'avantage d'être de niveau L2 et assez simple à suivre. Plein de ressources pour des TP, pour montrer comment s'implémente certaines notions. Rédigé par deux professeurs de l'Université de Rennes 1 en 2008!

[Autebert, Calculabilité] “Calculabilité et décidabilité : une introduction” 912,913,914,915,928

Un bon bouquin.

[Sipser] “Introduction to the theory of computation” 912,913,914,915,928

Une autre bonne référence pour les leçons de calculabilité, mais écrite en anglais. Prenez la dernière édition de 2013.

Logique et théorie des preuves 916,918,924

Concernent les leçons suivantes :

- **916** : Formules du calcul propositionnel : représentation, formes normales, satisfiabilité. Applications.
- **918** : Systèmes formels de preuve en logique du premier ordre : exemples.
- **924** : Théories et modèles en logique du premier ordre. Exemples.

[Mathieu Matthieu] “Logique pour l’informatique” 916,918,924

Un livre très récent, mais très recommandé, car il contient toute la logique de niveau agrégation, et bien plus. Il ne fait pas de SQL, mais un peu de théories des bases de données par contre.

[Lalement] “Logique, réduction, résolution” 916,918,924

Un bon bouquin qui traite de réécriture et d’unification (qui ne sont plus au programme), mais pas seulement puisqu’il donne des rappels sur la déduction naturelle, le théorème de Herbrand etc, qui seront utiles pour les leçons **916**, **918**, **924**. On trouvera notamment de bons exemples : fonctions récursives simples calculées par réécriture comme la factorielle ou Ackermann, les règles de dérivation formelles (II.2.1 p66), ou encore le fameux exemple de la théorie équationnelle des groupes (bien fait en V.3.3 p239).

[Cori1] et [Cori2] “Logique mathématique : cours et exercices corrigés”, Tomes 1 et 2 916,918,924

Deux excellents livres, à considérer plutôt comme un seul découpé en deux (tome 1 réédité en 2003, tome 2 en 2020). De nombreux exercices, des rappels de cours précis et concis, et des démonstrations plutôt claires, mais dont la longueur et la typographie un peu désuète pourront rebuter le néophyte (pour les vieilles éditions).

[RDavid] “Introduction à la logique : théorie de la démonstration” 916,918,924

Un bon complément aux Cori1 et Cori2! Excellente référence pour les développements de logique.

[Goubault] “Proof Theory and Automated Deduction” 916,918,924

Une bonne référence, même s’il n’est pas toujours facile à prendre en main. Attention à certaines preuves qui restent fausses, et qui sont irrattrapables (détails dans les preuves des théorèmes de Skolem et Herbrand pour le calcul des séquents du premier ordre LK_1 , par exemple). **Pas disponible** à la BU de l’ENS de Rennes, mais dans la malle d’agrég. (peut-être à la BU de Beaulieu?)

[Lassaigne] “Logique et complexité” 916,918,924

Peu de contenu utile pour le programme de l’agrég, mais peut éventuellement aider pour aller un peu hors du programme en théorie de la complexité, utile pour la fin du plan sur la leçon complexité. Une édition plus récente (2004) est disponible, en anglais (mais pas à Rennes, peut-être dans la malle d’agrég @Cachan).

Analyse lexicale et syntaxique 923

Concerne la **leçon 923** (Analyses lexicale et syntaxique : applications.).

[LegendreSchwarzentruber] “Compilation : analyse lexicale et syntaxique : du texte à sa structure en informatique” 923

Une excellente référence sur l’analyse lexicale et syntaxique (co-écrite par François Schwarzentruber un professeur de l’ENS de Rennes). Couvre tout ce qu’il faut inclure dans la leçon 923 et bien plus, avec de nombreux exemples.

Lambda-calcul 929

Concerne surtout la leçon **929** (Lambda-calcul pur comme modèle de calcul. Exemples) et un peu la leçon **930** (Sémantique des langages de programmation. Exemples). Il peut aussi être utile de mentionner le lambda-calcul comme un modèle de calcul équivalent aux fonctions primitives récursives donc pour la leçon **912** (Fonctions récursives primitives et non primitives. Exemples.) et aux machines de Turing donc pour la leçon **913** (Machines de Turing. Applications.).

[Krivine] “Lambda-calcul : types et modèles” 929

Un petit livre (1990) qui couvre tout le programme de la leçon Lambda-calcul. La partie sur les types est hors programme.

[Barendregt] “(The) lambda calculus : its syntax and semantics” 929

Un très bon livre en anglais sur le(s) lambda-calcul, qui va bien au delà du programme d'agrég de la leçon Lambda-calcul. Donne des preuves des principaux théorèmes qui peuvent être des développements pour cette leçon.

[Lassaigne2] “Logique et fondements de l'informatique : logique du 1er ordre, calculabilité et lambda-calcul” 929

Peu de contenu utile pour le programme de l'agreg, mais peut éventuellement aider pour aller un peu hors du programme en théorie de la complexité, utile pour la fin du plan sur la leçon complexité. Traite de lambda-calcul aussi.

Sémantique des langages de programmation 930

Couvrent la leçon **930** (Sémantique des langages de programmation : exemples.).

[Winskel] “The formal semantics of programming languages : an introduction” 930

Semble être une bonne référence (et la seule) pour la sémantique formelle des programmes, notamment la logique de Hoare et les preuves avec un invariant entre chaque lignes du programmes. Je crois me souvenir qu'il y a un exemple bien traité sur la fonction factorielle dans un langage simple impératif (IMP), qui peut faire un développement pour la leçon 927 (preuve programme) ou 930 (sémantique).

Bases de données 932

Pensez que les gros livres qui couvrent l'**Informatique pour tous** en prépa (programme d'après 2013) ont une partie entière sur les bases de données : définitions, notations formelles de l'algèbre relationnelle, exemples et illustrations, syntaxe du SQL, etc.

Couvrent la leçon **932** (Fondements des bases de données relationnelles.).

[Abiteboul] “The Foundations of Databases” 932

Probablement le meilleur livre pour préparer la leçon Base de données, même s'il est en anglais. Il est très théorique, et constituera une bonne référence pour aller loin dans la leçon.

Les livres suivants sont plus récents, je ne le connais pas encore.

[Gardarin] “Bases de données” 932

Un livre qui doit couvrir tout le programme pour la leçon Base de données. Il y a aussi un livre plus vieux, de 1999, mais utilisez la version la plus récente de 2003.

[Hainaut] “Bases de données : concepts, utilisation et développement” 932

Un point de vue beaucoup plus pratique. Le début est probablement utile pour la leçon Base de données, mais je pense que la majeure partie du livre est inutile.

[Meier] “Introduction pratique aux bases de données relationnelles” 932

Un point de vue beaucoup plus pratique. Le début est probablement utile pour la leçon Base de données, mais je pense que la majeure partie du livre est inutile.

Culture générale informatique

Ne concerne aucune leçon en particulier, mais se forger (ou plutôt se consolider) une bonne culture générale en informatique (sans négliger l'histoire de l'informatique, et ses grands succès récents) est une très bonne idée. Cela vous permettra de trouver des introductions, des conclusions ou des exemples et des illustrations intéressantes pour compléter vos soutenances de plans, développements et l'oral de modélisation.

[Chabert] “Histoire d’algorithmes : du caillou à la puce”

Surtout intéressant pour l'aspect historique de certains domaines de l'algorithmique. Notamment utile pour la méthode de Héron, la méthode de Gauss, etc.

[DowekEnseignement] “Introduction à la science informatique : pour les enseignants de la discipline en lycées”

Un très bon livre qui s'attaque à la question de comment bien enseigner l'informatique au niveau lycée.

[DowekCalcul] “(Les) métamorphoses du calcul : une étonnante histoire de mathématiques”

Un livre de culture générale sur l'histoire des mathématiques et du calcul.

[AbiteboulDowek] “Le temps des algorithmes, par Serge Abiteboul et Gilles Dowek”

Un excellent petit essai (grand public) sur la place croissante qu'occupent les algorithmes dans nos vies. Une lecture recommandée, qui peut aider à commencer des leçons d'algorithmique (ou autre) et des oraux de modélisation.

[Abiteboul] “Le bot qui murmurait à l'oreille de la vieille dame : et autres nouvelles numériques”

Un petit livre drôle et intéressant. A lire pour le plaisir.

[Tarissan] “Au coeur des réseaux : des sciences aux citoyens”

Un bon petit essai assez grand public sur les réseaux et l'informatique dans le monde contemporain.

Architecture des ordinateurs (modélisation)

Il y a depuis 2019 un point du programme spécifique à l'épreuve de modélisation d'option D :

- (a) **Éléments d'architecture des ordinateurs** : principaux composants et leurs interactions ; principes des langages assembleurs ;
- (b) **Représentation des nombres entiers et flottants** : ce point là notamment est TRÈS souvent le sujet d'étude de textes de modélisation !
- (c) **Éléments sur les systèmes d'exploitation** : systèmes de fichiers, processus, gestion de la mémoire.

Ces livres couvrent ces points du programme (et bien plus).

[Longchamp] “Introduction aux systèmes informatiques : architectures, composants, mise en œuvre”

Un bon livre.

[SchwarzArchi] “Architecture des ordinateurs”

Un bon gros livre dont les premiers chapitres sont à lire et à connaître (mémoire et unités de calculs, composants d'un ordinateur etc).

[CazesDelacroix] “Architecture des machines et des systèmes informatiques”

Un autre bon livre.

[Zanella] “Architecture et technologie des ordinateurs : cours et exercices corrigés”

Un autre bon livre.

Cryptographie (pas au programme)

Ce n'est pas au programme. mais ça peut être utile pour des développements à la frontière entre maths et info (leçons anneau $\mathbb{Z}/n\mathbb{Z}$ etc).

[Meunier] “Algèbre avec applications à l'algorithmique et à la cryptographie”

Un très bon bouquin, rappelle les bases sur le cours en algèbre mais va assez loin sur les applications (Diffie-Hellman, RSA, El-Gamal, codes correcteurs, Berlekamp, pseudo-inverse, FFT, et même Miller-Rabin).

[Menezes] “Handbook of applied cryptography”

Une excellente référence (en anglais) pour tout ce qui concerne la cryptographie. Un peu obscur et pas très clair sur les preuves, mais de bons schémas, des exercices et plein d'exemples (de tout, notamment Diffie-Hellman, RSA, ou El-Gamal).

Logique en lien avec l'unification et réécriture 919,920

PLUS AU PROGRAMME : les leçons **919, 920** sont sorties du programme.

[TermRewriting] “Term rewriting and all that” 919,920

THE reference (en anglais) pour tout ce qui touche à la réécriture, donc crucial pour les leçons **919** et **920**.