RÉVISIONS EXHAUSTIVES INFORMATIQUE MPI/MPI*

Algorithmes, Structures de Données, Théorie et Pratique

Résumé

*Ce document synthétise les algorithmes, concepts théoriques et techniques essentiels à maîtriser **parfaitement** pour les écrits. L'accent est mis sur la rapidité d'implémentation et la rigueur théorique.*

Table des matières

1 Algorithmes Éclairs (Objectif : 2 min)

À savoir recoder les yeux fermés en C ou OCaml.

Bases Arithmétiques et Tableaux

- Recherche du **minimum/maximum** d'un tableau.
- Algorithme d'**Euclide** (PGCD).
- **Exponentiation rapide** (itératif et récursif).
- Évaluation de polynôme (Méthode de **Horner**).
- **Décomposition en base** b (liste des chiffres).
- **Suite récurrente linéaire** (Fibonacci en $O(n)$ temps, $O(1)$ espace).
- **Insertion** dans une liste triée.
- Recherche par **dichotomie** (attention aux indices de fin).
- Produit matriciel naïf ($O(n^3)$).
- Miroir d'une liste en temps linéaire (OCaml, récursivité terminale).

Bases Arbres

- Calcul du **nombre de nœuds** et de la **hauteur** (arbre quelconque).
- **Parcours** : Préfixe, Infixe, Postfixe (DFS), Largeur (BFS).

2 Algorithmes Classiques (Objectif : 5 min)

2.1 Tris et Structures

Incontournables

- **Tri Rapide (Quick Sort)** : En place (partitionnement) et version fonctionnelle.
- **Tri Fusion (Merge Sort)** : Surtout en OCaml (diviser pour régner).
- **Tri par Tas (Heap Sort)** : Gestion d'un tas binaire.
- **Union-Find** : 'find' avec compression de chemin, 'union' par rang.
- **Pile/File** : Implémentation par liste chaînée (C) ou deux piles (OCaml).

2.2 Graphes

Parcours et Applications

- **BFS** (Largeur) : File. Plus court chemin en nombre d'arêtes.
- **DFS** (Profondeur) : Pile ou Récursif. Dates de début/fin.
- **Composantes Connexes** : Via parcours.
- **Tri Topologique** : Via DFS (date de fin décroissante) ou degrés entrants (Kahn).
- **Kosaraju** : Composantes Fortement Connexes (2 DFS).

3 Algorithmes Complexes (Principe et Application)

Savoir les dérouler sur un exemple papier et expliquer le principe.

3.1 IA et Apprentissage

- **Jeux** : Min-Max, Élagage Alpha-Beta, Attracteurs.
- **Apprentissage Supervisé** : k-plus proches voisins (k-NN), Arbres de décision (ID3).
- **Apprentissage Non-Supervisé** : k-moyennes (k-means), Classification Hiérarchique (CHA).

3.2 Graphes Avancés

Optimisation dans les Graphes

- **Dijkstra** : PCC (poids positifs). Tas binaire ou tableau dense.
- **Floyd-Warshall** : PCC toutes paires (Programmation Dynamique).
- **A*** : Heuristique (Dijkstra guidé).
- **Kruskal** / Prim : Arbre Couvrant Minimum (ACM).
- **Couplage Max** : Graphe biparti (Augmentation).

3.3 Texte et Automates

- **Recherche de motifs** : Naïf, Rabin-Karp (Hash), Boyer-Moore (Sauts), Knuth-Morris-Pratt (KMP).
- **Compression** : Huffman (Arbre optimal), LZW.
- **Automates** : Berry-Sethi (RegExp $\rightarrow$ Automate), Glushkov.

4 Méthodologie de la Preuve

Le trio gagnant

1. **Terminaison** : Exhiber un **variant** (entier positif strictement décroissant).
2. **Correction** : Exhiber un **invariant** de boucle (vrai avant, pendant, après).
3. **Complexité** :
   - Itératif : Somme des coûts.
   - Récursif : Équation de récurrence $T(n) = aT(n/b) + f(n)$ (Master Theorem).

- **Programmation Dynamique** : Sous-problèmes chevauchants, mémoïsation, reconstruction de solution. (Ex : Sac à dos, Distance d'édition, Plus longue sous-séquence).
- **Glouton** : Choix local optimal. (Preuve par échange).
- **NP-Complétude** : Savoir réduire un problème connu vers le problème cible (polynômialement).

5 Théorie Fondamentale

5.1 Logique

- **Propositionnelle** : Table de vérité, Satisfiabilité (SAT), Formes Normales (FNC/FND).
- **Premier Ordre** : Termes, Formules, Variables libres/liées, Substitution, Déduction Naturelle (Règles).

5.2 Langages et Automates

Automates Finis

- **Opérations** : Union, Concaténation, Étoile, Miroir, Complémentaire.
- **Algorithmes** : Déterminisation (Parties), Minimisation, Élimination ϵ.
- **Théorèmes** : Kleene (RegExp $\equiv$ Automate), Lemme de l'Étoile (Pumping Lemma pour montrer non-régulier).

- **Grammaires Hors-Contexte** : Arbres de dérivation, Ambiguïté.

5.3 Calculabilité

- **Indécidabilité** : Problème de l'Arrêt (Preuve par diagonalisation).
- **Réduction** : $A \leq B$ (A se réduit à B). Si A indécidable, B aussi.
- **Classes** : P (Polynomial), NP (Vérification Polynomiale), NP-Complet.

6 Systèmes, Concurrence et SQL

6.1 Concurrence

- **Concepts** : Thread, Race Condition, Deadlock, Famine.
- **Outils** : Mutex, Sémaphores, Variables Conditionnelles.
- **Problèmes** : Producteur-Consommateur, Lecteurs-Rédacteurs, Philosophes.

6.2 Bases de Données

SQL et Modélisation

- **Modélisation** : Entité-Association (Cardinalités 0,1 / 1,n / etc.), Schéma Relationnel (Clés primaires/étrangères).
- **Requêtes** : 'SELECT', 'FROM', 'WHERE', 'GROUP BY', 'HAVING', 'JOIN' (Inner, Left), Sous-requêtes.
- **Algèbre Relationnelle** : Sélection σ, Projection π, Jointure $\bowtie$.