

Révisions pour les vacances d'été

Récapitulatif Option Info MPSI

MPSI Lycée Pierre de Fermat

Ce document est un guide de révision pour les vacances, afin d'arriver bien préparés pour les cours d'option info en MP. Lisez le en entier au début des vacances, et déterminez les notions que vous pensez ne pas maîtriser sur le bout des doigts, histoire d'évaluer la quantité de travail que vous prendront vos révisions. Au moins deux semaines avant la rentrée, commencez les révisions : Revenez chapitre par chapitre sur le cours, et révisez comme vous en avez l'habitude (fiches, recopie du cours, exos, etc...), en repassant en priorité sur les notions qui vous posent le plus de problèmes. Profitez-en pour revenir sur les TP que vous n'avez pas fini.

1 OCaml

Il faut maîtriser toute la syntaxe de base d'OCaml, et être capable d'écrire du code correct sans ordinateur. Des points importants à revoir :

- Utilisation des `let`, différence entre un `let in local` et un `let sans in` qui étend le contexte global du programme.
- Définir un type somme avec plusieurs constructeurs, faire des `match-with` pour manipuler les éléments d'un type somme.
- Traits impératifs : boucles `for` et `while`, utilisation de référence lorsque l'on veut une variable mutable, tableaux.

De manière générale, les tableaux se manipulent avec des boucles, tandis que les listes se manipulent plutôt avec des fonctions récursives.

Fonctions classiques Vous n'avez pas besoin de connaître le nom de toutes les fonctions des modules standards d'OCaml, mais il faut connaître et pouvoir utiliser sans rappel les fonctions

`Array.length`, `Array.make`, `Array.make_matrix`.

Si vous connaissez quelques fonctions basiques du module `List`, comme `List.map`, `List.length`, `List.mem`, ça vous permettra de gagner en efficacité pour ne pas avoir à les recoder (sauf si le sujet le demande explicitement).

2 Étude de fonction

- Pouvoir justifier la terminaison d'une fonction récursive en exhibant une quantité positive entière et strictement décroissante (ce que vous avez pu appeler un variant en ITC).
- En lisant le code d'une fonction, pouvoir proposer une relation de récurrence vérifiée par la complexité.

- Connaître par coeur certaines complexités classiques :

Relation	Résultat	Algorithmes
$C_n = C_{n-1} + \mathcal{O}(1)$	$C_n = \mathcal{O}(n)$	maximum d'un tableau, recherche d'un élément
$C_n = C_{n-1} + \mathcal{O}(n)$	$C_n = \mathcal{O}(n^2)$	tri par insertion, par sélection
$C_n = C_{\frac{n}{2}} + \mathcal{O}(1)$	$C_n = \mathcal{O}(\log n)$	recherche par dichotomie
$C_n = C_{\frac{n}{2}} + \mathcal{O}(n)$	$C_n = \mathcal{O}(n \log n)$	tri fusion, tri rapide

Pouvoir dérouler une relation de récurrence sur la complexité (typiquement, pouvoir prouver les 4 cas précédents).

- En bonus, la récursivité terminale : pourquoi c'est utile, comment identifier une fonction récursive terminale, et comment écrire de telles fonctions à l'aide d'accumulateurs

3 Arbres

- Définitions basiques des arbres binaires, formules sur les arbres binaires stricts reliant le nombre de feuilles, de noeuds interne, et la hauteur.
- Parcours d'arbre récursif : préfixe, infixé, postfixé
- Parcours d'arbre en profondeur avec une pile, en largeur avec une file
- Arbres binaires de recherche : principe, implémentation des opérations en $\mathcal{O}(h)$ en ne descendant que d'un côté de l'arbre à chaque fois.

4 Structures de données

- Définition SDA / SDC
- SDA à connaître : pile, file, dictionnaire
- Pouvoir utiliser une structure de donnée si on vous donne l'interface (par exemple, savoir utiliser les fonctions du module Queue pour les files)

5 Logique propositionnelle

- Définitions du cours : formule, valuation, conséquence et équivalence logique, formule tautologique, satisfiable.
- Calcul propositionnel : simplifier une formule, tracer des tables de vérité.
- Problème SAT : pouvoir modéliser un problème en posant des variables propositionnelles, et en exprimant les contraintes comme des formules.
- Forme normale conjonctive, forme normale disjonctive, méthode pour trouver une FNC / FND à partir de la table de vérité.

6 Graphes

Bien que les graphes soient au programme d'ITC en première année, l'option info de deuxième année revient sur les parcours de graphes et présente d'autres algorithmes classiques. Revoyez :

- les définitions de base des graphes
- les représentations en mémoire par matrice ou liste d'adjacence
- les algorithmes de parcours en profondeur / largeur.