EXERCICES (pour les oraux informatisés de Centrale, TP et TIPE merci d'utiliser les feuilles spécifiques)

NOM DE L'ETUDIANT : GOTHIE										Concours 2024	ENS Paris-Saclay
Classe :	931	932	933	941	942	943	971	972			ENS LYON
EPREUVE	MATHS			S.I.							ENS ULM
	PHYSIQUE			LV							X
	CHIMIE			EPS							CENTRALE I pb ouvert
	INFO			FRANÇAIS							MINES
	Examinateur, date et heure : Lundi 8 juillet, 9h30										CC INP
	Durée de préparation :										TPE
	Durée de passage :										Autre:
	Note :										

Sujet de l'exercice : si vous faites un schéma, précisez s'il était fourni

27 min préparation, 27 min passage (+3 min à chq fois pour administratif + connexion ordis)

Exercice 1 [A]

Karatsuba pour des polynômes $P, Q \in \mathbb{R}_{2n-1}[X]$

(la formule $PQ = P_0Q_0 + X^n\left[(P_0+Q_0)(P_1+Q_1) - P_0Q_0 - P_1Q_1\right] + X^{2n}\left[P_1Q_1\right]$

où $P = P_0 + X^n P_1$ et $Q = Q_0 + X^n Q_1$

ainsi que des fonctions auxiliaires (leur spécification) étaient données)

1) Écrire en pseudo-code un algorithme qui calcule le produit de deux polynômes. On s'appuiera sur fonc° auxiliaires données

2) Donner la complexité [des fonctions auxiliaires] si on représente les polynômes par des tableaux de coefficients.

3) Justifier que la complexité temporelle de l'algo 1, pour des polynômes de degré [au plus ?] n vérifie :

$$C(n) \leq 3C\left(\frac{n}{2}\right) + Kn$$

4) Déterminer α tel que $C(n) = \mathcal{O}(n^\alpha)$. On se placera dans le cas $n = 2^p$

[D'autres questions]

Exercice B — Mots de Dyck — Programmation en C et compilation avec un Make-file.

- Code Fourni.
- Des définitions, exemples...

Résultat sur les nbs de Catalan admis et $C_n = \frac{(2n)!}{(n+1)!\,n!}$ donné.

1) Compléter le prototype `uint-64 catalan (int n)`
On pourra utiliser une fonc° auxiliaire.

2) Le programme va-t-il renvoyer le bon résultat si n est ≥ 10 ? Est-ce le cas ici ? Rq : la fonc° de test était déjà écrite.

3) Compléter le prototype `bool verification (string* mot)` qui indique si le mot est bien parenthésé ou non. Rq : fonction de test déjà écrits.

4) Déterminer la complexité de verificateur

pas sûre, j'ai un peu oublié.

5) Complexité d'un algorithme en force brute qui donne tous les mots bien parenthésés
[Des explications sur un algorithme de backtracking relatif aux mots de Dyck]

6) Compléter prototype de la fonction dyck qui suit cet algorithme

[2 ou 3 autres questions].

Indications et remarques faites par l'examinateur (en particulier sur la gestion du temps) :

* au bout de 15 min, l'examinateur m'a prévenue, mais j'étais libre de poursuivre sur l'exercice de mon choix.

* 5 min avant la fin j'avais fini de présenter tout ce que j'avais fait. Il m'a laissé le choix de l'exercice à poursuivre. J'ai pris type-B. Comme je n'avais pas le temps de coder, il m'a demandé, en guise de QG, d'expliquer à l'oral ce que faisait l'algo en question.

Commentaires de l'élève : Il était sympa et menait l'oral de manière bienveillante.