ÉCOLE POLYTECHNIQUE
ENS ULM
ENS PARIS-SACLAY
ENS LYON
ENS RENNES
ESPCI

CONCOURS D'ADMISSION 2026

Composition d'Informatique A, Filières MP et MPI

1 Le sujet

Le sujet portait sur les bases de données de vecteurs, une structure de données qui permet d'insérer des vecteurs à d dimensions et d'identifier les plus proches voisins d'un vecteur en requête passé en argument.

La partie 1 portait sur les algorithmes de **reconnaissance automatique d'extraits audio** (utilisés dans la vie de tous les jours), à partir de diverses représentations de leur spectrogramme, i.e. diagramme temps-fréquence [1]. Certains élèves ont eu des difficultés avec la représentation en binaire des entiers (I.3 à I.5). Par exemple, pour la question I.3, au lieu d'utiliser l'instruction `lsl` rappelée dans le sujet, beaucoup ont réécrit une fonction puissance. La question I.8 portant sur le calcul efficace à l'aide d'une file, plus difficile, a été merveilleusement traitée par certaines copies. Certains candidats croient toutefois que `Queue.length` est en temps $O(n)$ alors qu'on rappelle dans le sujet que cette fonction s'exécute en temps $O(1)$. L'argument attendu : « chaque élément a été inséré une fois et défilé une fois donc la complexité est amortie linéaire », a été retrouvé dans plusieurs copies. Pour la question de modélisation de I.9, ce qui était important n'était pas que l'écart soit petit, mais que l'histogramme soit concentré autour d'une seule valeur, indiquant la cohérence temporelle des correspondances.

La partie 2 portait sur les **listes à sauts** (*skip lists*) [2]. Il s'agit d'une structure de données probabiliste qui permet d'insérer et de rechercher des éléments (la suppression n'était pas traitée dans le sujet) à l'aide de successeurs à plusieurs étages. La fin du sujet portait sur une implémentation de bases de données de vecteurs à base de **graphes hiérarchiques navigables** [3], une généralisation des listes à sauts. C'est finalement la compréhension du concept même de liste à saut qui a été discriminante dès les premières questions de la partie (II.1 et II.3). Pour la question II.1, certaines copies commencent la recherche à `init_level` alors que pour cette question il faut commencer à 0. Pour la question II.2, les copies ont proposé toutes sortes de valeurs exotiques mais la bonne était np^ℓ. Pour la question II.3, aller vers la droite ou le bas permet d'obtenir une complexité $O(n + L)$. Trop de copies recommençaient la recherche tout à gauche à chaque étage ou bien confondaient `init_level` avec `max_level`. Les questions II.4 et II.5 consistaient à analyser la complexité en moyenne de la recherche d'élément. Bien que difficiles, ces questions ont été parfois bien traitées dans de rares copies. Certaines copies ont même proposé une preuve plus élégante que l'article original de William Pugh [2]. La question sur l'insertion (II.6), bien que difficile, a été relativement bien comprise par la plupart des candidats, surtout II.6.a qui portait sur le nombre de successeurs à modifier au maximum, où nous avons accepté les réponses L et $2L$. Enfin, la question II.7 pouvait être comprise comme : parcourir une seule arête par étage, ou bien faire un parcours glouton à chaque étage jusqu'à tomber dans un minimum local (la solution attendue). Nous avons accepté les deux solutions, et la solution attendue s'est retrouvée dans la plupart des copies. Beaucoup ont remarqué que la distance décroissait strictement, ainsi il n'était pas nécessaire de marquer les nœuds.

La partie 3 portait enfin sur l'algorithme des k moyennes, ainsi qu'à la quantification de produit [4], une méthode pour réduire la complexité des opérations de recherche de plus proche voisin pour des vecteurs de grande dimension. Il s'agit de l'implémentation la plus populaire des bases de données de vecteurs utilisés dans l'industrie aujourd'hui (pour la génération augmentée par récupération RAG dans les grands modèles de langue LLM par exemple), car elle est aisément parallélisable. Elle a été relativement bien traitée dans la plupart des copies : les questions III.1 à III.6, sauf éventuellement III.3 et III.5, sont une application directe du cours, donc III.4 et III.6 auraient pu être mieux traitées. À la question III.5, nous souhaitions faire référence au cas où un centroïde n'est affecté à aucun point, mais nous avons compté juste pour les copies ayant interprété différemment la question (centroïde « non mis à jour » pouvant signifier « centroïde inchangé »). Les questions III.7 et III.8 portaient enfin sur la quantification de produit et sa complexité et permettaient de comprendre l'intérêt de cette dernière partie. Quelques copies ont proposé des codes élégants tirant parti de la programmation fonctionnelle OCaml pour implémenter la quantification de produit (III.7).

2 Remarques générales et statistiques par question

À part certaines copies difficilement lisibles ou écrites trop petit, les correcteurs ont été globalement positivement surpris par la présentation des copies ainsi que la clarté de la justification, même lorsqu'ils n'en demandaient pas tant : le sujet employait subtilement parfois la locution « Donner », parfois « Justifier », mais dans l'ensemble, tout a été abondamment justifié par les candidats. Finalement, le sujet a eu un grand ambitus, de sa note la plus grave à sa note la plus élevée.

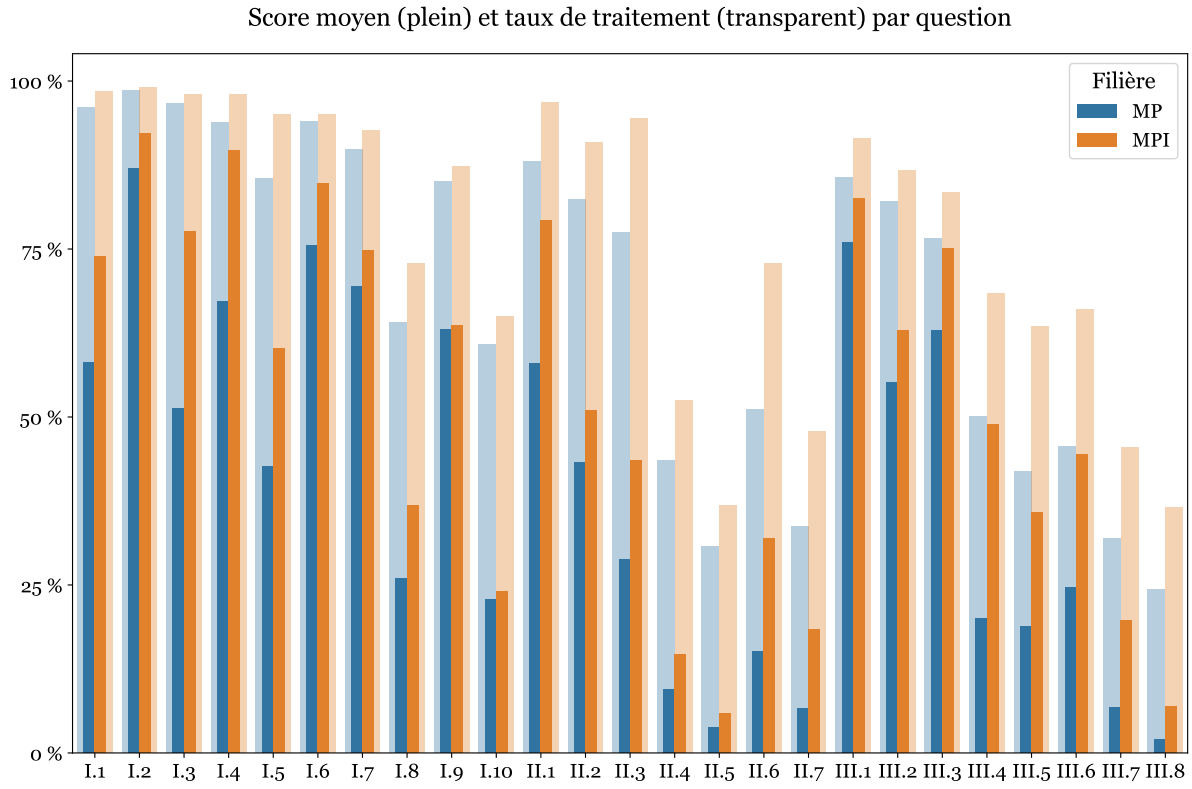


Figure 1 – Score moyen (non traité vaut zéro) et taux de traitement par question

On voit à la Figure 1 que les MPI ont plus traité et mieux traité que les MP **chaque** question du sujet, surtout I.1 (complexité de la recherche de motif dans un diagramme temps fréquence

Table 1 – Taux de traitement et score moyen si traité pour chaque question et filière MP et MPI

pourcentages	I.1	I.2	I.3	I.4	I.5	I.6	I.7	I.8	I.9	I.10
taux traitement MP	96	99	97	94	85	94	90	64	85	61
score moyen si traité MP	60	88	53	72	50	80	77	41	74	38
taux traitement MPI	99	99	98	98	95	95	93	73	87	65
score moyen si traité MPI	75	93	79	92	63	89	81	50	73	37

pourcentages	II.1	II.2	II.3	II.4	II.5	II.6	II.7
taux traitement MP	88	82	78	44	31	51	34
score moyen si traité MP	66	53	37	22	12	30	20
taux traitement MPI	97	91	94	53	37	73	48
score moyen si traité MPI	82	56	46	28	16	44	38

pourcentages	III.1	III.2	III.3	III.4	III.5	III.6	III.7	III.8
taux traitement MP	86	82	77	50	42	46	32	24
score moyen si traité MP	89	67	82	40	45	54	21	8
taux traitement MPI	91	87	83	68	63	66	45	37
score moyen si traité MPI	90	72	90	71	57	67	43	19

en $O(DTF)$), I.3 à I.5 (représentation en binaire des entiers), II.1 (taille d'une skip list), II.3 (recherche dans une skip list), II.6 (insertion dans une skip list), II.7 (parcours d'un graphe hiérarchique), III.4 à III.8 (k-moyennes et quantification de produit) ; pour ces dernières, probablement par rapidité. Notez que la Figure 1 indique le score moyen en supposant que non traité vaut zéro tandis que la Table 1 indique le score moyen si la question a été traitée. Par exemple, pour la question I.8, 64 % des MP ont traité cette question et ont eu en moyenne 41 % des points (cf. Table 1), soit un score moyen à cette question pour tous les MP de 26 % (en bleu dans la Figure 1).

Références

[1] Avery Li-Chun Wang. "An Industrial-Strength Audio Search Algorithm". In: *4th International Symposium on Music Information Retrieval (ISMIR)*. Washington, DC, pp. 7–13. URL: https://github.com/MichaelCStrauss/shazam-python/blob/main/Wang03-shazam.pdf.

[2] William Pugh. "Skip lists: a probabilistic alternative to balanced trees". In: *Communications of the ACM* 33.6 (1990), pp. 668–676. URL: https://g-trees.github.io/g_trees/assets/references/pugh1990skip.pdf.

[3] Yury A. Malkov and Dmitry A. Yashunin. "Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs". In: *IEEE transactions on pattern analysis and machine intelligence* 42.4 (2018), pp. 824–836. URL: https://arxiv.org/pdf/1603.09320.

[4] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. "Product quantization for nearest neighbor search". In: *IEEE transactions on pattern analysis and machine intelligence* 33.1 (2010), pp. 117–128. URL: https://inria.hal.science/inria-00514462/document.