

Composition d'Informatique A (XULSR)

Filière MP spécialité Info

1 L'épreuve

Dans ce sujet, on s'intéresse au problème de la différentiation algorithmique. Il s'agit de calculer efficacement des différentielles d'expressions mathématiques vues comme des composées de fonctions élémentaires. C'est l'une des pierres fondatrices du calcul formel, ainsi que de la descente de gradient dans les réseaux de neurones. Dans la première partie, on introduit des fonctions utiles dans la suite pour multiplier des matrices. Dans la deuxième partie, on s'intéresse à la multiplication d'une suite finie de matrices réelles. Il est notamment demandé aux candidats de montrer que l'ordre dans lequel on multiplie les matrices influence fortement la complexité, puis de développer un algorithme pour calculer un ordre optimal. Dans la troisième partie, on étudie deux manières de calculer la différentiation de la composition d'une liste de fonctions et d'implémenter des algorithmes correspondants en utilisant la deuxième partie. Dans la quatrième partie, on s'intéresse au cas plus général d'une fonction représentée par un graphe, ce qui nécessite d'ajuster les algorithmes précédents. Dans la cinquième partie, largement indépendante, on s'intéresse aux triangulations de polygones et aux liens qui existent avec la complexité de la multiplication d'un nombre fini de matrices. Pour obtenir la note maximale, il n'était pas nécessaire de traiter l'intégralité du sujet.

La moyenne des 960 candidats français est de 9,63/20 avec un écart-type de 3,66.

La moyenne des 67 candidats de nationalité étrangère est de 8,49/20 avec un écart-type de 4,54.

2 Remarques générales

Trop de copies s'avèrent difficiles à lire pour les correcteurs. Les candidats qui produisent des réponses raturées, mal indentées ou encore non structurées en paragraphe perdent nécessairement des points malgré tous les efforts de décodage des correcteurs. Les centres d'examen fournissent des feuilles de brouillon aux candidats. On rappelle que pour être valide une preuve de programme doit suivre avec précision la structure du programme. Les preuves par induction, et notamment les preuves par récurrence, ne sont pas maîtrisées par un grand nombre de candidats. Outre l'oubli ou le mauvais choix du cas de base, le choix de la quantité ou la structure sur laquelle on effectue l'induction est souvent hasardeux. Pour les questions sur la correction d'un algorithme, on attend autre chose qu'une simple paraphrase de cet algorithme.

3 Commentaire détaillé

Pour chaque question, sont indiqués entre crochets le pourcentage de copies ayant traité la question et le pourcentage de copies ayant obtenu la totalité des points.

Question I.1 [99%, 86%] Il s’agissait là d’une question très facile, surtout destinée à être utilisée par la suite.

Question I.2 [99%, 81%] Il s’agissait là d’une question facile, mais de trop nombreux candidats ne savent pas ce qu’est un produit scalaire.

Question I.3 [99%, 89%] Il s’agissait là d’une question facile.

Question I.4 [99%, 71%] Il s’agissait là d’une question facile qui nécessitait surtout de maîtriser les boucles et d’utiliser les bonnes bornes.

Question II.1 [99%, 40%] De trop nombreux candidats pensent que le coût pour effectuer deux multiplications de matrice est le produit des coûts pour effectuer chacune d’elles. Par ailleurs, beaucoup de candidats ont conclu que “l’ordre d’évaluation dépend de la taille des matrices”, ce qui ne veut rien dire puisque l’ordre d’évaluation dépend de l’algorithme.

Question II.2 [99%, 84%] Il s’agissait là d’une question facile.

Question II.3 [98%, 57%] La multiplication de matrices n’étant pas commutative, utiliser une fonction récursive triviale sur le modèle de `muld` ne marche pas. De même, `mulg` n’est pas la composée de `muld` et de `List.rev`. Par ailleurs, beaucoup de candidats effectuent des doubles renversements de liste pour en extraire le dernier élément, ce qui est correct mais pas très rassurant.

Question II.4 [82%, 9%] Beaucoup de candidats n’ont considéré que les arbres de multiplication qui sont des peignes (c’est-à-dire que chaque noeud interne est nécessairement relié à une feuille). Cette erreur a généralement impacté tout le reste de la partie II.

Question II.5 [71%, 8%] Beaucoup de candidats ont rempli la matrice par programmation dynamique, mais sans se soucier de l’ordre de remplissage. En particulier, remplir la matrice par lignes croissantes ne donne pas le bon résultat. Remplir la matrice par mémoïsation permettait de contourner cette difficulté.

Question II.6 [39%, 13%] Certains candidats, plutôt que de d’abord calculer la décomposition du produit optimal, ont directement cherché à calculer le produit optimal. Ils ont ainsi calculé tous les produits $B_i \times \dots \times B_j$ sans le vouloir, ce qui n’a pas du tout la bonne complexité.

Question III.1 [95%, 59%] Cette question ne présentait pas de difficulté particulière.

Question III.2 [93%, 87%] Il s’agissait là d’une question facile.

Question III.3 [71%, 13%] Beaucoup de candidats ont écrit des codes inutilement compliqués et incorrects. Peu de preuves sont convaincantes car à base d’arguments vagues ou ne suivant pas la structure du code proposé.

Question III.4 [49%, 12%] Peu de candidats ont correctement traité cette question. De nombreuses copies oublient totalement de prendre en compte le coût de la multiplication, ce qui est évidemment problématique. Un nombre important de copies ne calculent pas correctement la somme géométrique obtenue.

Question III.5 [49%, 19%] Cette question ne présentait pas de difficulté particulière, mais beaucoup de candidats ont cherché à y répondre de la même façon qu'à la question III.3, ce qui a compliqué les choses inutilement.

Question III.6 [32%, 6%] Cette question se traitait de la même façon que la question III.4 et a donc souffert des mêmes problèmes.

Question III.7 [31%, 9%] Il s'agissait en fait d'une question facile, mais peu de candidats l'ont remarqué.

Question IV.1 [69%, 18%] Beaucoup de candidats n'ont pas compris l'énoncé et en particulier que la différentielle était à évaluer sur le vecteur $(1, 1)$. Ils se sont donc engagés dans des calculs trop compliqués.

Question IV.2 [48%, 29%] Cette question était facile, mais beaucoup de candidats ont extrait les premières valeurs du tableau plutôt que celles dont les indices ont été passés en argument. Autrement dit, ils n'ont utilisé que la taille du tableau d'indices plutôt que son contenu.

Question IV.3 [20%, 4%] Il s'agissait là d'une question assez difficile car il fallait bien comprendre ce qui précédait. Beaucoup de candidats n'ont pas initialisé correctement le tableau et ont fait des erreurs dans les calculs des différentielles.

Question IV.4 [12%, 4%] Cette question était mal formulée et admettait des réponses triviales.

Question V.1 [51%, 3%] Cette question a été traitée par de nombreux candidats pensant qu'il s'agissait d'une question facile. Toutefois, de trop nombreuses copies ne posent pas de récurrence, ne précisent pas sur quoi porte la récurrence, voire donnent une "preuve" à base d'exemples. Les correcteurs apprécient les exemples qui peuvent aider à comprendre le raisonnement mais ceux-ci ne peuvent être suffisants.

Question V.2 [28%, 12%] Cette question, pourtant plutôt facile, a été mal traitée par beaucoup des candidats qui ont fait des erreurs d'indices ou se sont contentés d'explications vagues.

Question V.3 [35%, 5%] De trop nombreux candidats ont tenté de mesurer l'orientation de trois points en utilisant le signe d'un produit scalaire alors que l'énoncé mentionnait explicitement l'utilisation d'un produit vectoriel. Par ailleurs, beaucoup de candidats ont supposé que ce signe devait forcément être positif pour un polygone convexe. Il y a aussi beaucoup de candidats qui ont calculé le produit vectoriel des sommets consécutifs plutôt que celui des arêtes consécutives. Enfin, excepté dans une poignée de copies, tous les algorithmes proposés sont incorrects dans le cas dégénéré où un des angles est plat ; cela n'a pas été sanctionné.

Question V.4 [42%, 5%] Beaucoup de candidats se sont contentés de prouver qu'ils peuvent effectuer une triangulation en $n - 3$ arêtes plutôt que de prouver qu'une triangulation arbitraire produit $n - 3$ arêtes.

Question V.5 [27%, 13%] Beaucoup de candidats ont mis dans leur triangulation deux arêtes internes en trop, celles qui correspondent à des sommets successifs.

Question V.6 [16%, 4%] Cette question a été traitée par peu de candidats mais plutôt bien traitée quand c'était le cas. La difficulté résidait dans le traitement correct du cas où une des liste contenait l'indice 0.

Question V.7 [4%, 0%] Cette question était quasiment impossible.