

# Informatique

## Présentation du sujet

Le sujet porte sur le thème de la recherche de plus court chemin, appliqué au cas d'un système embarqué sur le robot Spirit, où une optimisation des déplacements entre points de mesure permet d'économiser de l'énergie.

La première partie s'intéresse à l'extraction de points d'intérêt où le robot doit réaliser des mesures, soit de façon aléatoire pour des besoins de tests de l'algorithme, soit par analyse d'image, ou encore par extraction des informations dans une base de données.

La deuxième partie propose une première approche du calcul du chemin entre les points d'intérêt par l'algorithme du plus proche voisin. Une comparaison de complexité est faite entre le traitement « par force brute », c'est-à-dire en déterminant la longueur de la totalité des chemins possibles et l'algorithme du plus proche voisin, bien plus rapide. La dernière question permet néanmoins de montrer que l'algorithme ne détermine pas nécessairement le chemin le plus court.

La troisième partie aborde une seconde approche, par un algorithme génétique, mieux à même de trouver un chemin plus court, mais pour lequel il faut trouver un critère d'arrêt adapté.

## Analyse globale des résultats

Le sujet est de longueur raisonnable pour le temps imparti. De nombreux candidats abordent la totalité du sujet.

À nouveau cette année, le jury se réjouit du niveau satisfaisant des copies. Le langage est bien maîtrisé et permet de traduire les solutions aux questions sans difficultés. Seule une petite proportion des candidats (moins de 5%) rend une copie vide ou contenant des programmes sans aucune cohérence, ne sachant pas écrire une boucle, réaliser une simple affectation correctement ou discerner le nombre de dimensions d'un tableau. Ces copies conduisent à quelques notes très faibles et demeurent une énigme pour le jury, après trois semestres de cours et de travaux dirigés en informatique. Il est possible que ces copies soient celles de candidats n'ayant pas suivi d'enseignement d'informatique de classe préparatoire.

Les petites erreurs syntaxiques n'ont pas été retenues par le jury comme un élément discriminatoire, dans la mesure où elles ne cachent pas des erreurs de fond. Les réponses pertinentes d'un point de vue algorithmique sont valorisées.

Certaines copies proposent des programmes particulièrement élégants et concis et reflètent un vrai recul sur les différentes stratégies de programmation. Ces copies ont été valorisées.

## Commentaires sur les réponses apportées et conseils aux futurs candidats

Au regard des copies évaluées, le jury propose aux futurs candidats de prêter attention aux remarques suivantes.

L'indentation en Python délimite les blocs d'instructions et doit apparaître clairement dans la rédaction. Toute présentation claire est bienvenue ; bien souvent, un trait vertical marquant l'alignement du bloc d'instruction est suffisant.

L'initialisation d'une variable dans une boucle ou hors de la boucle n'a pas les mêmes conséquences pour l'algorithme.

Le nombre d'itérations d'une boucle doit être bien réfléchi pour s'assurer que les indices des éléments d'une liste appelée dans la boucle sont bien définis. L'instruction `range(n)` parcourt `n` itérations indicées de 0 à `n-1`.

La concision et l'élégance des programmes sont appréciées dans l'évaluation. Les candidats qui réinvestissent les fonctions déjà codées sont valorisés par rapport à ceux qui recopient les lignes de code équivalentes. Bien souvent, une condition booléenne bien choisie permet d'éviter de longues listes de conditions aux instructions identiques.

Des noms de variables explicites aident à la compréhension du code. De trop nombreux candidats utilisent des noms de variables non significatifs (`a`, `b`...) ce qui nuit à la compréhension du programme. La clarté du programme (en particulier le choix des noms de variables) ainsi que la présence de commentaires opportuns sont prises en compte dans l'évaluation.

Lors d'un commentaire sur une complexité, une justification chiffrée minimale est attendue : un programme de complexité exponentielle pourrait être utilisable pour peu que le nombre d'opérations soit faible au regard de la puissance de calcul d'un micro-processeur actuel.

L'ordre des questions importe. Prendre soin de rédiger les réponses aux questions en respectant leur ordre dans le sujet.

La qualité d'expression (l'orthographe notamment) et la qualité visuelle de présentation relèvent des compétences de communication indispensables à un candidat à une école d'ingénieur. Le correcteur n'attribue les points qu'aux éléments de réponse qu'il parvient à lire et à comprendre. Les copies obscures et difficiles à comprendre sont pénalisées.

Les variables utilisées dans une fonction doivent être définies dans cette fonction ou être explicitement définies comme variables globales (soit par le sujet, soit par le candidat). Beaucoup de candidats ont utilisé la variable `n` sans la définir, ce qui a soulevé une ambiguïté, `n` pouvant être le nombre de points d'intérêt incluant la position initiale du robot ou pas.

Les candidats sont invités à bien lire l'annexe contenant certaines fonctions utiles pour traiter le sujet.

## Première partie

Les points d'intérêt choisis au hasard devaient être distincts. Certains candidats ne vérifient pas cette contrainte dans leur programme. Le calcul des distances devait prendre en compte (dans la dernière colonne) les distances à la position courante du robot. De nombreuses erreurs sur les indices ont été relevées, que ce soit sur le nombre d'itérations ou les indices des éléments du tableau. Certains candidats confondent encore l'affectation (`=`) et le test d'égalité (`==`).

Les explications sur la fonction `F1` se limitent parfois à paraphraser les lignes de code. Il était attendu une description de la tâche globale réalisée et une interprétation du résultat renvoyé.

Les deux premières questions de bases de données sont très souvent abordées et bien réussies. C'est moins le cas pour les trois questions suivantes. Les questions 3 et 5 faisaient intervenir des jointures. La question 4 nécessitait de prendre des initiatives en définissant le nombre de bit de codage des entiers pour en déduire la surface maximale d'exploration enregistrable.

## Deuxième partie

Cette partie débute avec deux fonctions relativement simples et utiles pour la suite. La longueur du chemin est souvent bien calculée, lorsque la position courante du robot est prise en compte. En revanche, la normalisation fait souvent l'objet de programmes décousus. L'erreur la plus fréquente étant d'engager une boucle sur la longueur du chemin tout en supprimant des éléments de ce même chemin, ce qui ne manque pas de conduire à un débordement et à des décalages d'indices.

La plupart des candidats trouvent le nombre de chemins possibles mais un minimum de justification était attendu. Pour conclure, il fallait non seulement réaliser une application numérique pour déterminer l'ordre de grandeur du nombre d'itérations à prévoir, mais aussi pour comparer ce nombre aux puissances de calcul des processeurs actuels.

L'algorithme du plus proche voisin était probablement le plus complexe à élaborer, sachant qu'il convenait de s'assurer de ne pas repasser sur un point déjà visité. Les réponses ont souvent conduit à des programmes sans queue ni tête, difficilement évaluables par le correcteur en l'absence de commentaires. Le calcul de complexité devait prendre en compte la complexité d'une instruction telle que `if k not in L`.

Beaucoup de candidats ont trouvé un exemple où l'algorithme ne fournit pas le plus court chemin. Un schéma était souvent plus clair qu'un long discours à cette question.

## Troisième partie

Les fonctions `créer_population` et `réduire` distinguent les candidats qui savent utiliser judicieusement les fonctions mises à disposition et ceux qui tentent de se débrouiller sans. Les questions suivantes n'ont pas posé beaucoup de difficultés aux candidats attentifs aux indices. Une grande partie des candidats est à l'aise avec la manipulation des listes pour effectuer des concaténations et du slicing.

La construction de la fonction principale `algo_génétiq`, réutilisant bon nombre des fonctions précédentes, est bien réussie. L'argumentation sur le critère d'arrêt est souvent pertinente lorsque la question est abordée.

## Conclusion

Le sujet aborde une large partie du programme d'informatique commune. Le choix d'un sujet s'appuyant sur un thème courant en informatique assure une cohérence avec la formation d'ingénieur. Cette approche sera reconduite sur des problématiques de simulation ou d'algorithmique en informatique, à partir du programme des trois semestres d'informatique.

Les bons résultats à cette épreuve montrent que les étudiants, soutenus par leurs professeurs, ont acquis des compétences affirmées en informatique. Le jury encourage les futurs candidats à travailler l'informatique en alliant réflexion sur feuille de papier et mise en œuvre des algorithmes sur ordinateur.