

Option Informatique

Présentation du sujet

Le sujet 2017 de l'option informatique traite des mots synchronisants, très utiles notamment pour rétablir un état particulier d'une machine à partir d'une situation quelconque. Après une étude théorique, la mise en œuvre des algorithmes demandés nécessite la manipulation de vecteurs, de listes, de structures construites. Un peu de logique et quelques études de complexité complètent le sujet. Le problème est découpé en parties indépendantes, mais il est nécessaire de bien lire les hypothèses et d'analyser correctement comment les données sont représentées. Cette année encore, le choix d'un texte de longueur raisonnable permet aux candidats d'aborder l'ensemble du problème.

Analyse globale des résultats

Le sujet a été globalement compris. Les meilleurs candidats ont pu traiter le problème dans son ensemble. Il reste quelques copies très faibles, de candidats visiblement non investis dans la matière. Les signatures des fonctions `Caml` étaient imposées ; elles ont été bien respectées, mais nous avons constaté de grandes difficultés sur le respect des types des structures de données ainsi que sur les filtrages. De manière générale, nous avons observé des confusions importantes entre `Caml` et `Python` et une rigueur insuffisante dans la syntaxe `Caml`. Il reste également des points très mal compris, par exemple sur l'usage de `référence` même pour des constantes, ou utilisées sans déréférencement. Et, malheureusement, toujours les mêmes travers signalés tous les ans. Beaucoup de candidats compliquent le sujet en n'utilisant pas les éléments précédemment définis, omettent de numérotter les questions, écrivent des codes complexes sans aucun commentaire, ou sur plusieurs pages, ou encore en multipliant inutilement le nombre de fonctions auxiliaires rendant la lecture très difficile, oubliant se faisant qu'ils doivent être lisibles par des humains et qu'il est peu fréquent que la réponse à une seule question nécessite obligatoirement 4 ou 5 fonctions auxiliaires. Les calculs de complexité sont trop souvent pas ou mal justifiés. Cela dit, le niveau moyen est satisfaisant et la présentation souvent agréable, avec des changements de page opportun ou de la couleur, ainsi que des indentations correctes.

Commentaires sur les réponses apportées et conseils aux futurs candidats

Les principales difficultés observées dans la première partie tiennent au fait que les candidats lisent mal le sujet. De plus, nous devons rappeler qu'un minimum de justification / démonstration des réponses est attendu, même sans que la demande soit explicite dans le texte. Quand un argument n'est pas nécessaire, question I.C. par exemple, le texte le précise. Il est nécessaire de respecter les types donnés par le sujet : dans certaines copies les listes deviennent des vecteurs. Plus surprenant, l'entier `n_etats` a été transformé en `liste`, mais il a aussi été considéré comme `mutable`. Même si ce n'est pas explicitement indiqué, quand on teste si un mot est, ou non, synchronisant, il semble normal de s'arrêter dès qu'on vérifie qu'il ne l'est pas ; beaucoup de candidats calculent tout, et même parfois deux fois, avant de conclure. Il convient également de faire attention aux bornes dans les boucles : si `V` est un vecteur de longueur `n`, `V.(n)` n'est pas défini. Enfin il reste des

incompréhensions importantes sur ce qu'est un automate déterministe. À noter que, comme pour les codes, le jury n'imagine pas donner des expressions rationnelles sur une demi-page.

Comme son titre l'indique, la deuxième partie porte sur des algorithmes classiques. La difficulté réside dans l'utilisation de types particuliers définis dans le texte et qu'il fallait respecter. C'est ce qu'ont fait un grand nombre de candidats bien sûr, mais même les questions II.A.1 et II.A.2 se sont révélées difficiles. Trop de réponses proposées incohérentes entre elles, mauvaise maîtrise des enregistrements, bien que la syntaxe soit rappelée, implémentation d'une pile et non d'une file, confusion fréquente entre la liste et le vecteur, confusion explicite quand les candidats suggèrent que la complexité de `vect_length` est $O(n)$. Pour la suite, on ne peut que rappeler aux candidats que, en effet, il arrive qu'un sujet reprenne des éléments très certainement traités en cours, mais qu'il ne suffit pas de le constater en guise de justification ; plusieurs candidats se limitent à dire qu'il s'agit « d'un parcours en largeur » pour répondre aux questions. Plusieurs fois dans le problème, des candidats indiquent qu'ils « pourraient faire une récurrence », annoncée comme « évidente », mais sans aucune indication de la propriété, ni même de la variable, sur laquelle pourrait porter cette récurrence.

Ces éléments de réponses trop partiels se retrouvent dans les justifications sur les dernières parties. Trop souvent les candidats ne prennent pas le temps de poser des démonstrations précises. Attention également à respecter les types demandés, par exemple « vecteur de listes » pour la machine.

Dernier conseil aux candidats, invariablement le même, ne pas oublier que ce sont des humains qui corrigent. Comme nous l'avons déjà signalé, cela oblige à écrire des codes clairs, mais également à utiliser des notations courantes ou explicitées, et bien sûr à conserver celles du texte quand elles sont données.

Conclusion

Le temps de formation en informatique est limité dans nos classes et le jury mesure la difficulté imposée par l'apprentissage simultané de deux langages dont la philosophie est très différente. De ce fait, il faut absolument pratiquer sur machine pour acquérir des réflexes. La lisibilité des codes, la précision des démonstrations, la conformité des réponses avec les questions posées sont primordiales. Pour cela, il faut avoir des idées claires sur les éléments du programme et sur les bases du langage et bien sûr avoir une bonne capacité d'adaptation. Il faut également prendre le temps de bien lire et comprendre les implications des hypothèses et des indications qui sont données dans le texte.

Reste que le niveau global des candidats est satisfaisant. Un nombre important de copie est même d'un excellent niveau, alors qu'il est indéniablement difficile d'écrire des codes sans l'aide d'un compilateur. Le jury félicite les candidats qui s'investissent ainsi dans la discipline.