

## 4.2 Informatique option MP

### 4.2.1 Généralités

Le sujet s'intéresse à l'analyse et à la programmation de méthodes de correction des erreurs dans un mot. Il est composé de trois parties : une première partie s'intéresse à la distance d'édition entre deux mots, une seconde partie à une représentation s'appuyant sur une structure d'arbres et de dictionnaires et une troisième partie traite de la manipulation de scripts de transformations.

27 questions composent le sujet.

Dans la première partie, il s'agit de mettre en œuvre des techniques usuelles d'analyse d'une fonction récursive puis de mémoïsation. Nous avons constaté que peu de candidats savent manipuler la mémoïsation.

Dans la seconde partie, le candidat doit s'approprier la structure élaborée des objets décrits (les tries). Il peut alors réfléchir convenablement aux fonctions demandées. Beaucoup de candidats ont éprouvé de grandes difficultés. Ils n'ont pas compris la structure donnée.

La troisième partie est très peu abordée convenablement.

Les candidats ont abordé de façon équilibrée les questions de programmation ainsi que les questions portant sur des démonstrations.

### 4.2.2 Analyse de forme

Les programmes présentés par les candidats respectent les règles d'indentation avec des retours à la ligne facilitant la lecture du code. Plusieurs copies restent malgré tout mal présentées voire même illisibles pour le correcteur, sans indentation, avec de grosses ratures, des renvois avec des flèches en bas de page, etc. Ces copies sont très difficiles à interpréter.

Plusieurs compositions utilisent des fonctions auxiliaires, ce qui est une bonne chose pour décomposer un programme. Il est recommandé d'utiliser des noms significatifs à l'image de ce que font les concepteurs des sujets. Un commentaire additionnel permet de mieux évaluer la compréhension du sujet ou de la question par les candidats.

Les correcteurs souhaitent mentionner la présence de preuves d'équivalence souvent imprécises lorsqu'elles sont conduites par une succession d'équivalences. Peu de compositions font la démonstration dans les deux sens.

Enfin, plusieurs copies renferment encore des expressions de la forme « c'est évident », « il est clair que ... », « trivial ». Ces expressions ne sont pas acceptées lorsque la question exige une justification et/ou bien une démonstration qui doit être convaincante et rigoureusement établie. De la même façon, on constate de façon généralisée un manque de rigueur dans la rédaction des preuves (ici notamment par récurrence ou induction). Beaucoup de candidats se contentent d'un descriptif approximatif et ne vérifient pas l'ensemble des hypothèses ou tirent des conclusions hâtives sans rappeler (ni vérifier, par extension) les conditions nécessaires à la validité du raisonnement.

### 4.2.3 Analyse par question

- Q1** - Quelques erreurs surprenantes de typage de la fonction. Les justifications sont souvent approximatives. En particulier,  $a \leq b$  n'implique pas que  $a$  et  $b$  sont des entiers.
- Q2** - Une preuve par induction ou récurrence évite toute approche approximative. Peu de preuves complètes. D'autres sont trop approximatives.
- Q3** - Même remarque.
- Q4** - Il y a souvent confusion avec la complexité temporelle. On note une difficulté à argumenter la complexité annoncée.
- Q5** - Bien traitée en général.
- Q6** - Trop peu de bonnes réponses. Les candidats semblent déconcertés par la recherche d'une formule simple.
- Q7** - Bien traitée en général même si beaucoup de candidats utilisent une approche très impérative qui nuit à la lisibilité.
- Q8** - On constate dans cette question que beaucoup de candidats ne savent pas mettre en œuvre un algorithme de programmation dynamique et restent cantonnés à une approche récursive.
- Q9** - Bien traitée en général quand la question précédente est faite.
- Q10** - Peu traitée.
- Q11** - Bien traitée en général.
- Q12** - On constate une méconnaissance des différentes implémentations possibles d'un dictionnaire et de la complexité de l'opération d'insertion. Peu de réponses correctes.
- Q13** - **Q14** - **Q15** - **Q16** - Ces questions nécessitent une bonne compréhension de la structure introduite dans l'énoncé. Elles nécessitent aussi une certaine rigueur dans la manipulation des types. Les candidats qui ont fait l'effort de compréhension et de rigueur donnent souvent des codes appropriés.
- Q17** - La notion de structure persistante semble peu maîtrisée.
- Q18** - Peu traitée.
- Q19** - Beaucoup d'argumentations fantaisistes.
- Q20** - Peu de bonnes réponses. Une confusion avec la structure usuelle d'automates en est sans doute la cause : peu de candidats ont perçu le caractère très spécifique ici.
- Q21** - Peu traitée.
- Q22** - Peu traitée.
- Q23** - Beaucoup d'argumentations superficielles.
- Q24** - **Q25** - **Q26** - **Q27** - Peu traitées.