

Devoir surveillé n° 1 — barème

I Trois exercices [/ 21]

Exercice 1 : Prévoir le résultat [/ 6]

- I.1) [/ 5] Question globalement peu réussie, et rares sont ceux qui expliquent leurs réponses.
 I.2) [/ 1] Un manque de fantaisie qui a déçu le correcteur.

Exercice 2 : Tri insertion [/ 5]

C'était une question de cours, ou tout comme, et sans surprise. Le code est globalement correctement écrit, à une ou deux fantaisies près.

- I.3) [/ 2]
 I.4) [/ 1]
 I.5) [/ 2] La justification de la complexité est rarement parfaite. Attention aux conclusions hâtives cachant maladroitement des calculs (simples, ici). Certains n'ont pas saisi que le meilleur et le pire cas doivent être considérés pour un n donné : on ne peut pas citer le cas de la liste vide comme cas le plus favorable.

Exercice 3 : Écriture en base b [/ 10]

- I.6) [/ 1] Calculs basiques mais sur lesquels certains ...
 I.7) [/ 1] ... arrivent à se tromper quand même. Relire ses calculs !
 I.8) [/ 4] Beaucoup de problèmes dans cette question, avec un certain nombre de copies dans lesquelles on ajoute un élément en queue de liste, ce qui trahit une mauvaise compréhension de la structure de liste chaînée. Peu nombreux sont ceux qui pensent à typer la fonction (ce qui rapportait un quart des points de la question).
 I.9) [/ 4] Attention, il ne faut surtout pas appeler `rev` à chaque appel récursif sous peine d'obtenir une complexité quadratique plutôt que linéaire. Question rarement réussie (et fonction non typée en général). Faire le calcul sans renverser la liste n'est pas raisonnable à cause du coût des exponentiations.

II Représentation d'ensembles avec des intervalles [/ 22]

- II.1) [/ 1] De nombreux élèves répondent `int * int -> int` ce qui est peut-être plutôt dû à un problème d'énoncé : on supposait implicitement que les paires étaient des paires d'entiers dans tout le problème. De nombreux autres élèves trouvent `'a * 'a -> 'a`. Il y a aussi confusion du type et de la réponse de CAML, ainsi beaucoup de réponses donnent comme `type : ('a * 'b) -> 'a = <fun>`, même si ce n'est pas très grave.
 II.2) [/ 2] Cette question peut être très pénible à corriger si elle n'est pas simplement exposée.
 II.3) [/ 1] On attendait une réponse simple.
 II.4) [/ 2] On peut utiliser `min : 'a -> 'a -> 'a` et `max : 'a -> 'a -> 'a`. Mais de trop nombreuses copies écrivent `min (a b)`.
 II.5) [/ 1] Question très mal traitée. La réciproque, puisqu'il s'agissait d'établir une équivalence, n'a presque jamais été évoquée. Les hypothèses « bien formés » et « disjoints » étaient nécessaires.
 II.6) [/ 2] Question très simple, mais presque jamais réussie. Il suffisait juste de montrer que les intervalles étaient disjoints *deux à deux*, mais il fallait le faire.

- II.7) [/ 4] Question bien réussie. Quelques rares élèves « pensent » que $x \in [a, b] \iff (x = a) \vee (x = b)$. Il ne faut pas oublier de parler de la complexité.
- II.8) [/ 3] Question plus difficile, très bien traitée par ceux qui l'abordent sérieusement. Il fallait bien lire et comprendre l'énoncé.
- II.9) [/ 2] La liste vide ne peut pas être le meilleur cas pour une donnée *de taille* n ! Il fallait bien dire que *fusion* et *disjoints* étaient en $\Theta(1)$.
- II.10) [/ 4] Lorsqu'elle est abordée, cette question a été traitée avec succès.

III Autour du tri rapide [/ 36]

- III.1) [/ 1] De trop nombreux élèves trouvent $(\text{'a list} \rightarrow \text{'a}) \rightarrow (\text{'a list} * \text{'a} * \text{'a list})$ au lieu de $\text{'a list} \rightarrow \text{'a} \rightarrow (\text{'a list} * \text{'a} * \text{'a list})$ et ce n'est pas la même chose.
- III.2) [/ 2] La réponse à cette question est longue et il faut trouver un compromis pour ne pas y perdre trop de temps, en choisissant une présentation des calculs récursifs claire et efficace. Mieux vaut une bonne indentation que trop de verbiage.
- III.3) [/ 2] On attend ici une explication de comment mais aussi *pourquoi* ce programme trie effectivement une liste. La plupart des réponses sont satisfaisantes.
- III.4) [/ 1] Réponses en général satisfaisantes. L'étape *régner* consiste à effectuer les appels récursifs sur les sous-problèmes pour les résoudre, pas à dérécursifier jusqu'aux cas de base.
- III.5) [/ 1] Question sans difficulté. Il faut bien dire qu'une proposition du type $\forall i \in \mathcal{O}, \mathcal{P}(i)$ est toujours vraie et non pas qu'elle est « impossible ».
- III.6) [/ 1] Question triviale, il suffisait de recopier l'énoncé, ce qui a été fait.
- III.7) [/ 3] Question plus délicate peu abordée.
- III.8) [/ 1] Inutile de grappiller, seuls ceux ayant répondu aux questions précédentes pouvaient avoir le point.
- III.9) [/ 6] Question non abordée.
- III.10) [/ 2] Très peu abordée, mais correctement.
- III.11) [/ 1] Très peu abordée, mais correctement.
- III.12) [/ 0] Question très simple, mais on attendait $T(0) = T(1) = 0$ et non un $\Theta(1)$ puisqu'ici on veut compter *exactement* les opérations de comparaisons qui sont les seules opérations élémentaires.
- III.13) [/ 1] Très peu abordée, mais correctement.
- III.14) [/ 1] Question non abordée.
- III.15) [/ 1] Question non abordée.
- III.16) [/ 3] Question non abordée.
- III.17) [/ 3] Question non abordée.
- III.18) [/ 3] Question non abordée.
- III.19) [/ 3] Question non abordée.