

**Composition d'Informatique (4 heures), Filière MP
(XULC)**

Rapport de MM. Charles-Edmond BICHOT, Jean-Christophe FILLIÂTRE, Jérôme LEROUX et Mme Assia MAHBOUBI, correcteurs.

1. L'épreuve

Il s'agissait dans ce problème d'étudier plusieurs algorithmes de transmission d'informations dans des arbres. L'épreuve alternait des questions de programmation et des questions visant à établir des résultats d'optimalité pour certains algorithmes de transmission.

La **partie I** introduisait les arbres binomiaux, une famille classique d'arbres contenant 2^k nœuds et de hauteur k . Les candidats devaient répondre à des questions élémentaires sur les arbres binomiaux et programmer leur construction. Cette partie introduisait également une version tronquée de l'arbre binomial, dans le but de faciliter une question ultérieure. La **partie II** avait pour but le calcul du diamètre d'un arbre, par une méthode originale consistant à effectuer un changement de racine. Cette partie contenait une majorité de questions de programmation. La **partie III** étudiait le problème de la diffusion dans les arbres, c'est-à-dire de la transmission d'un message depuis la racine à tous les noeuds. On considérait tout d'abord le cas particulier des arbres binomiaux, puis le cas général. Enfin la **partie IV** étudiait le problème de l'échange total dans les arbres, où chaque nœud possède un message qu'il doit transmettre à tous les autres nœuds. Là encore, le sujet commençait par l'étude du cas particulier des arbres binomiaux, avant de proposer un algorithme d'échange total s'appliquant à tous les arbres.

2. Remarques générales

Il s'agissait de la première édition d'une épreuve écrite commune pour les concours d'admission de l'École Polytechnique et des Écoles Normales Supérieures.

Les notes des 781 candidats français se répartissent selon le tableau suivant, avec une moyenne de 8,83 et un écart-type de 3,42. Le nombre de notes éliminatoires, c'est-à-dire inférieures à 2, parmi les candidats français est de 18.

$0 \leq N < 4$	44	5,6 %
$4 \leq N < 8$	285	36,5 %
$8 \leq N < 12$	313	40,1 %
$12 \leq N < 16$	115	14,7 %
$16 \leq N \leq 20$	24	3,1 %
Total	781	100 %
Nombre de copies : 781		
Note moyenne : 8,83		
Écart-type : 3,42		

Concernant les questions de programmation, presque toutes les fonctions demandées pouvaient être écrites en moins de 10 lignes, indépendamment du langage de programmation choisi. Les candidats doivent être conscients du fait qu'une réponse longue doit être expliquée en détail et que très souvent un programme très long contient un grand nombre d'erreurs.

Pour obtenir la note maximale, il était nécessaire de traiter le problème en entier.

3. Commentaire détaillé

Pour chaque question, sont indiqués entre crochets la moyenne de la question sur 20 ainsi que le pourcentage de candidats ayant obtenu la note 0 (que la question ait été traitée ou non). De nombreuses questions étaient notées de manière binaire ou ternaire, sur la base de 0/100% ou 0/60/100%.

Partie I

Question 1 [18,4 - 2%]. Cette question élémentaire a pour but d'aider les candidats à visualiser la structure d'un arbre binomial. Les principales erreurs proviennent d'une mauvaise lecture de l'énoncé menant à un étiquetage incorrect (doublons, lettres,...) ou absent.

Question 2 [16,3 - 8%]. Cette question est correctement traitée par la majorité des candidats. Là encore quelques candidats lisent l'énoncé trop vite et oublient de répondre à la deuxième partie de la question.

Question 3 [18,5 - 7%]. Cette question est correctement traitée par la majorité des candidats. Cependant, certains candidats perdent du temps en donnant une réponse ex-

cessivement détaillée. Il n'est pas nécessaire d'être verbeux pour être rigoureux.

Question 4 [16,2 - 9%]. Cette question est le plus souvent correctement traitée. Attention toutefois à la confusion entre les primitives `do_list` et `map` du langage Caml.

Question 5 [12,8 - 13%]. Les deux principales difficultés de cette question sont de respecter la complexité demandée par l'énoncé et d'assurer une numérotation correcte, en particulier sans doublons. Certains candidats donnent une réponse correcte en faisant un calcul explicite de 2^k , soit via une fonction auxiliaire de calcul en $\log(k)$, soit dans le cas du langage Caml en utilisant la primitive `ls1`. D'autres candidats donnent une réponse correcte et élégante en évitant le calcul explicite de la puissance, comme dans le programme suivant :

```
let rec bin = function
| 0 -> (1, [])
| n -> let (r, l) = bin (n - 1) in
         let (t, copie_l) = copie r (r, l) in
         (t, (r, l) : : copie_l);;
```

ou bien de façon encore plus concise :

```
let rec bin = function
| 0 -> N (1, [])
| k -> let N (a, l) as t = bin (k-1) in N (2*a, copie (a-1) t : : l);;
```

Les erreurs de complexité sont le plus souvent imputables à la présence de trois appels récursifs, par exemple deux appels à `bin (k - 1)` plus un appel à la fonction `copie`.

Question 6 [12,0 - 4%]. La plupart des candidats justifient correctement la valeur de la profondeur. La majorité des candidats donne une valeur correcte pour la longueur maximale d'un chemin mais par contre très peu en donnent une preuve rigoureuse. Beaucoup de candidats utilisent en effet sans le justifier le fait qu'un chemin de longueur maximale dans un arbre binomial passe forcément par la racine.

Question 7 [14,6 - 18%]. Cette question est bien traitée dans l'ensemble ; les candidats maîtrisent visiblement la rédaction de ce type de raisonnement. Néanmoins, quelques candidats donnent le résultat sans preuve.

Question 8 [12,5 - 12%]. La plupart des candidats dessinent correctement C_{16} . En revanche la justification de la structure d'arbre est moins bien traitée, voire absente. Il s'agit de montrer que $1 \geq pere(i) < i$, et l'inégalité de gauche est souvent oubliée.

Question 9 [2,8 - 79%]. Cette question est traitée, quel que soit le langage de programmation choisi par les candidats. À l'exception de quelques rares copies qui proposent une

solution concise et élégante, les programmes proposés sont incorrects.

Partie II

Question 10 [16,3 - 8%]. Cette question est généralement bien traitée. L'erreur la plus fréquente consiste à donner une solution en complexité quadratique alors que l'énoncé demande un algorithme linéaire en le nombre de noeuds.

Question 11 [11,2 - 25%]. Un grand nombre de candidats donnent un programme de complexité quadratique en le nombre de noeuds alors que l'énoncé demande une complexité linéaire. Cette erreur provient le plus souvent de l'utilisation dans l'appel récursif de la fonction profondeur programmée à la question précédente.

Question 12 [9,0 - 38%]. Cette question est bien traitée dans l'ensemble. Quelques candidats tombent néanmoins dans l'écueil de la complexité quadratique. Attention là encore, il faut bien lire l'énoncé et choisir en conséquence les valeurs calculées pour les cas de bases : certains candidats écrivent des programmes qui calculent le chemin renversé, ou bien qui présentent deux occurrences du premier élément, ou bien encore une liste vide quand l'argument est la racine.

Question 13 [3,6 - 72%]. Il s'agit d'une question de programmation difficile, peu traitée et mal le plus souvent. Pour ce type de question, nous recommandons aux candidats de commenter les solutions proposées, soit sous la forme de commentaires dans les programmes, soit en décrivant brièvement l'algorithme implémenté, ces deux suggestions n'étant pas exclusives.

Question 14 [1,6 - 89%]. Très peu de candidats répondent de façon rigoureuse à cette question, qui donne lieu à de grosses fautes de raisonnement. Par exemple, certains candidats font l'hypothèse que r_1 est un noeud externe qui est aussi une extrémité d'un chemin de longueur le diamètre de T , ce qui n'est pas conforme à l'énoncé.

Question 15 [14,0 - 29%]. Cette question très simple est généralement bien traitée.

Question 16 [3,1 - 66%]. Cette question est mal traitée dans la plupart des copies : les candidats confondent souvent arbre de départ et arbre pivoté, et ne traitent le plus souvent que la première partie de la question.

Partie III

Question 17 [8,9 - 33%] et 18 [9,0 - 35%]. La plupart des candidats qui traitent ces questions ont donné les bonnes valeurs. En revanche la justification est moins souvent convaincante, alors qu'il s'agit dans les deux cas d'un argument élémentaire. Attention encore une fois à l'étourderie : certains candidats lisent mal l'énoncé et inversent les deux questions.

Question 19 [11,7 - 41%]. Il était assez simple de remarquer que l'existence d'un chemin de longueur k donne la borne inférieure attendue. Certains candidats remarquent qu'en k étapes il y a au plus 2^k nœuds informés et en déduisent que comme $|Bk| = 2^k$, k est une borne inférieure, ce qui constitue également une réponse correcte. En revanche, d'autre abusent dans cette question des preuves par intimidation du type « Il est évident que...».

Question 20 [3,2 - 77%] et 21 [3,2 - 81%]. Certains candidats proposent d'essayer toutes les diffusions possibles pour trouver celle qui réalise le meilleur temps. Cependant la plupart des candidats qui traitent cette question comprennent l'énoncé de façon moins naïve et proposent une approche récursive. On ne demandait pas ici de connaître la complexité optimale du tri, néanmoins la complexité de la solution proposée est rarement étudiée. Certains candidats essaient d'utiliser la profondeur pour construire l'algorithme et donnent des solutions incorrectes. Curieusement, quelques candidats n'arrivent pas à décrire un algorithme correct à la question 20 mais répondent correctement à la question 21.

Question 22 [6,3 - 58%]. Les candidats qui traitent la question donnent le plus souvent la valeur correcte mais rarement sa justification. Attention à l'écueil du nombre d'intervalles entre n poteaux : certains candidats donnent une justification correcte mais une valeur incorrecte de n au lieu de $n - 1$.

Question 23 [3,8 - 67%]. Cette question est rarement bien traitée, en particulier la valeur donnée est souvent mal justifiée. Certains candidats expliquent toutefois que l'on atteint la borne inférieure en privant B_n de certaines de ses feuilles, ce qui constituait une réponse correcte.

Partie IV

Question 24 [3,4 - 75%]. Les candidats qui ont traité cette question ont en général décrit l'idée générale de l'algorithme mais les justifications données sont souvent approximatives.

Question 25 [4,4 - 78%]. Cette question est bien traitée quand elle est traitée.

Question 26 [0,3 - 93%]. Il s'agissait d'une question difficile, qui demandait une rédaction soignée. Très peu de candidats y ont répondu et les solutions proposées sont souvent erronées.

Question 27 [4,6 - 77%]. Les quelques candidats qui ont répondu à cette question ont le plus souvent donné une réponse correcte.

Question 28 [0,8 - 93%]. Les candidats qui ont répondu à cette question ont le plus souvent proposé des programmes corrects, ayant la complexité requise.

Question 29 [0,2 - 99%]. Très peu de candidats ont correctement traité cette question.