N° d'anonymat:

Année Universitaire 2022-2023
Session 1 de printemps

Parcours/Étape: L3 Informatique **Code UE:** 4TIN602U
Épreuve: Techniques Algorithmiques et Programmation
Date: 28/04/2023 **Heure:** 9h00 **Durée:** 1h30
Documents: une seule feuille A4 recto-verso autorisée.
Épreuve de M. Cyril Gavoille

université de BORDEAUX

Collège Sciences et Technologie

RÉPONDRE DIRECTEMENT SUR LE SUJET
QUI EST À RENDRE DANS LA FEUILLE DOUBLE D'EXAMEN

Question de cours

Question 1. *Parmi les notions suivantes, quelles sont celles qui ont été abordées en CM ou en TD ? [Cochez la ou les cases correspondantes.]*

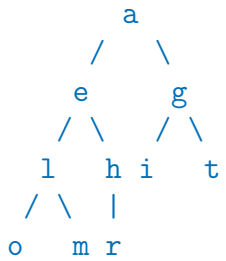
- ☐ *le Master Theorem*
- ☐ *les algorithmes de Deep Learning*
- ☐ *les algorithmes distribués*
- ☐ *la compilation*
- ☐ *l'indécidabilité*
- ☐ *la programmation dynamique*
- ☐ *les heuristiques*
- ☐ *les algorithmes parallèles*
- ☐ *les algorithmes probabilistes*
- ☐ *les tables de hachage*
- ☐ *la logique temporelle*
- ☐ *la programmation par contraintes*

Solution. **[6 pts]** Master Theorem, l'indécidabilité, la programmation dynamique, les heuristiques, les algorithmes probabilistes, les tables hachage (-1 pts par erreur).

Tas minimum de caractères alphabétiques

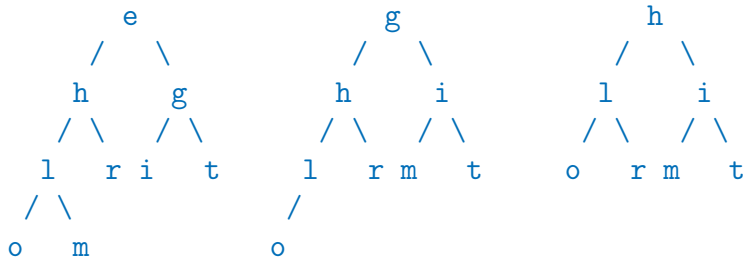
Question 2. *Donnez le tas résultant de l'insertion dans un tas minimum supposé vide au départ des 10 lettres du mot* `algorithme` *ajoutées une par une en le lisant de gauche à droite. On trie les lettres selon l'ordre alphabétique. Par exemple :* `g`$<$`r` *et* `m`$>$`e`.

Solution. **[3 pts]**



Question 3. *Appliquer trois fois l'opération de suppression du minimum au tas de la question précédente. Donner le tas obtenu après l'exécution de chacune opérations.*

Solution. **[3 pts]**



Question 4. *Quelles sont dans l'ordre les trois lettres ainsi supprimées ?*

SOLUTION. **[2 pts]** a, e, g

Minimum local dans un rectangle

On appelle *rectangle* toute grille rectangulaire R de dimensions $L \times C$ où chaque case $R[i][j]$ contient un nombre avec $i \in \{1, \ldots, L\}$ et $j \in \{1, \ldots, C\}$. Par convention on supposera que i représente le numéro de ligne et j le numéro de colonne. On note par $R[l_1..l_2][c_1..c_2]$ le sous-rectangle composé de tous les éléments $R[i][j]$ avec $i \in \{l_1, \ldots, l_2\}$ et $j \in \{c_1, \ldots, c_2\}$. Ainsi $R = R[1..L][1..C]$.

Voici l'exemple d'un rectangle $R[1..3][1..7]$.

	1	*2*	*3*	*4*	*5*	*6*	*7*
1	19	63	28	63	84	65	69
2	77	73	51	51	27	40	23
3	62	73	35	65	27	88	20

MINIMUM LOCAL est le problème qui consiste à trouver dans un rectangle R la position d'un *minimum local*, c'est-à-dire la position (i, j) d'un élément $R[i][j]$ plus petit ou égal à tous ses éléments voisins dans R. Les éléments voisins de $R[i][j]$ sont $R[i][j-1]$, $R[i][j+1]$, $R[i-1][j]$ et $R[i+1][j]$ s'ils existent. Certaines positions $(i \pm 1, j \pm 1)$ pouvant être en dehors de R, certains de ces éléments peuvent ne pas exister. La taille de R, notée n, est le nombre total d'éléments, c'est-à-dire $n = LC$.

Question 5. *Trouver tous les minimum locaux de l'exemple ci-dessus. [Encerclez-les directement sur l'exemple du sujet.]*

SOLUTION. **[3 pts]**

19	63	**28**	63	84	65	69
77	73	51	51	**27**	40	23
62	73	**35**	65	**27**	88	**20**

-1 pts par erreur.

Question 6. *Montrez que tout rectangle contient toujours au moins un minimum local.*

SOLUTION. **[3 pts]** Un minimum absolu, qui existe toujours, est nécessairement un minimum local.

Pour résoudre le problème MINIMUM LOCAL, on propose l'algorithme GRADIENT qui consiste à partir d'une position arbitraire du rectangle. Puis tant que l'élément de la position courante n'est pas un minimum local on se déplace vers un élément voisin strictement inférieur.

Question 7. *Montrez que la complexité de l'algorithme* GRADIENT *n'est pas plus que* $O(n)$.

SOLUTION. **[3 pts]** On ne peut pas passer deux fois pas un même élément car l'algorithme GRADIENT génère une suite d'éléments strictement décroissant. Il visite donc au plus une fois tous les éléments du rectangle, soit n.

Question 8. *Construisez un rectangle où l'algorithme* GRADIENT *a une complexité* $\Omega(n)$.

SOLUTION. **[2 pts]** On peut considérer une suite décroissante d'éléments rangés dans un rectangle ligne $R[1..1][1..n]$ avec $(1,1)$ comme premier choix de position. Un exemple fini, avec $n = 4$ par exemple, ne répond pas à la question.

Un *minimum global* est un élément de R qui est plus petit ou égal à tous les autres.

Question 9. *Montrez que le problème de la recherche d'un minimum global dans un rectangle a pour complexité* $\Theta(n)$.

SOLUTION. **[3 pts]** Tout algorithme qui examine au plus $n-1$ cases peut rater le plus petit élément. Donc la complexité du problème est au moins $\Omega(n)$. C'est au plus $O(n)$ avec un algorithme de parcours linéaire. La complexité est donc en $\Theta(n)$.

On propose une approche « diviser-pour-régner » pour résoudre le problème MINIMUM LOCAL. La *coupe médiane* d'un rectangle R est une ligne ou une colonne telle que sa suppression coupe R en deux sous-rectangles chacun de taille deux fois plus petite ou moins. Le choix de prendre une ligne ou une colonne se fait selon la taille d'une ligne ou d'une colonne : on choisit celle de plus petite taille.

Par exemple, si $R = R[1..3][1..7]$ est un rectangle 3×7, alors la colonne centrale $M = R[1..3][4..4]$ découpe R en deux sous-rectangles qui sont $A = R[1..3][1..3]$ et $B = R[1..3][5..7]$. Ici M est de taille 3, ce qui est plus petit que la taille d'une ligne qui vaut 7. De plus la taille de R est 21, alors que les tailles de A et de B valent 9.

L'algorithme DICHOTOMIQUE consiste :

(1) à découper R en deux sous-rectangles A et B obtenus en supprimant la coupe médiane ;
(2) à chercher la position (i, j) du minimum global de la coupe médiane ;
(3) si $R[i][j]$ est un minimum local pour R on renvoie (i, j), sinon on recommence récursivement dans le sous-rectangle A ou B contenant un voisin de (i, j) strictement plus petit.

Question 10. *Appliquez l'algorithme* DICHOTOMIQUE *à l'exemple initial* $R[1..3][1..7]$. *Donnez la position et la valeur du minimum trouvé.*

SOLUTION. **[2 pts]** Il y a deux découpages possibles, les deux menant au minimum local $R[3][7] = 20$.

On suppose qu'on dispose d'une fonction `C` donnant une implémentation de l'algorithme DICHOTOMIQUE, et dont le prototype est :

```
position Dicho(double **R, int l1, int l2, int c1, int c2);
```

où `position` est une structure permettant de décrire un couple d'indices. Avec cette fonction, il suffit donc d'appeler `Dicho(R,1,3,1,7)` pour répondre à la question précédente. Pour simplifier, on supposera que les indices `0` de `R` ne sont pas utilisés.

Question 11. *Donnez l'arbre des appels pour* `Dicho(R,1,3,1,7)`. *[Plusieurs arbres sont possibles. N'en donnez qu'un seul.]*

SOLUTION. **[3 pts]**
$(R, 1, 3, 1, 7) \to (R, 1, 3, 5, 7) \to (R, 1, 3, 7, 7)$
ou

$(R, 1, 3, 1, 7) \to (R, 1, 3, 5, 7) \to (R, 3, 3, 5, 7)$.

Question 12. *Pourquoi la technique de mémorisation ne semble pas intéressante pour l'algorithme* DICHOTOMIQUE *?*

SOLUTION. **[2 pts]** Car l'arbre des appels n'a pas de nœuds qui se répètent, donc pas de calcul inutile à mémoriser. En effet, l'arbre est toujours un chemin et donc s'il y avait un nœud qui se répètent la fonction bouclerait.

On s'intéresse à la complexité en temps $T(n)$ de l'algorithme DICHOTOMIQUE appliqué à un rectangle de taille n. Pour cela, on remarque qu'en jouant sur les indices de R, dans les étapes (1) et (3) de l'algorithme, il n'est pas nécessaire de créer les sous-rectangles A et B. Ainsi l'étape la plus couteuse est l'étape (2).

Question 13. *Montrez que pour tout* $L, C \geqslant 0$, $\min(L, C) \leqslant \sqrt{LC}$.

SOLUTION. **[2 pts]** On a $\min(L, C) = \sqrt{\min(L, C)^2}$, ce qui implique $\min(L, C) \leqslant \sqrt{\min(L, C) \cdot \max(L, C)} = \sqrt{LC}$.

Question 14. *Donnez une équation de récurrence pour* $T(n)$. *Justifiez.*

SOLUTION. **[3 pts]** $T(n) = O(\sqrt{n}) + T(\lfloor n/2 \rfloor)$. Le terme $O(\sqrt{n})$ s'explique par le fait qu'il faut un temps $O(\min(L, C))$ pour trouver le minimum global du médian, et on a vu précédemment que $\min(L, C) \leqslant \sqrt{LC} = \sqrt{n}$.

Question 15. *En déduire la complexité de l'algorithme* DICHOTOMIQUE. *[Aide : Utilisez le Master Theorem sur* $T(n)$ *ou le fait que* $\sum_{i \geqslant 0} q^i < \frac{1}{1-q}$ *pour* $q \in]0, 1[$.*]*

SOLUTION. **[2 pts]** D'après le Master Theorem avec $a = 1, b = 2, c = 0$, on a $\lambda = \log_b(a) = 0$ et $f(n) = \Theta(\sqrt{n})$. Comme $\sqrt{n} = \Omega(n^{\lambda+\epsilon}) = \Omega(n^\epsilon)$, c'est le 3e cas du Master Theorem qui s'applique. Il faut cependant vérifier que $a \cdot f(n/b + c) \leqslant q \cdot f(n)$ pour une certaine constante $q < 1$. C'est bien le cas, car $a \cdot f(n/b + c) = \sqrt{n/2} = 1/\sqrt{2} \cdot \sqrt{n} \leqslant q \cdot f(n)$ avec $q = 1/\sqrt{2} < 1$. Et donc la solution est $T(n) = \Theta(f(n)) = \Theta(\sqrt{n}\,)$.

Question 16. *Même question dans le cas où* R *est de dimension* $1 \times n$.

SOLUTION. **[2 pts]** Pour un rectangle $1 \times n$, l'algorithme DICHOTOMIQUE (qui revient dans ce cas à faire une dichotomie simple), mène à la récurrence $T(n) = O(1) + T(n/2)$. Et donc $T(n) = O(\log n)$ d'après le Master Theorem.

Un étudiant propose une variante plus rapide pour l'algorithme DICHOTOMIQUE où l'étape (2) est remplacée par (2') chercher récursivement la position (i, j) du minimum local de la coupe médiane.

Question 17. *Montrez alors que la complexité* $T(n)$ *de cette variante vérifie la récurrence* $T(n) = O(\log n) + T(n/2)$. *Quelle est alors la complexité de cette variante ?*

SOLUTION. **[2 pts]** D'après la question précédente, la recherche de l'étape (2) prend un temps $O(\log n)$, et donc, à cause de l'étape (3), $T(n)$ vérifie l'équation $T(n) = O(\log n) + T(n/2)$. Malheureusement, c'est un cas où le Master Theorem ne s'applique pas car $\lambda = 0$ ($a = 0, b = 2, c = 0$) et $f(n) = \log n$, et on ne tombe dans aucun des trois cas. Cependant, en déroulant $k = \lceil \log_b n \rceil = \lceil \log_2 n \rceil$ fois la récurrence pour $T(n)$, on obtient :

$$\begin{aligned} & \log(n/2^0) + \log(n/2^1) + \log(n/2^2) + \cdots + \log(n/2^k) \\ & (k+1)\log n - (0 + 1 + 2 + 3 + \cdots + k) \\ = \; & (k+1)\log n - k(k+1)/2 \leqslant (k+1)\log n - (\log n)(k+1)/2 \\ \leqslant \; & \frac{1}{2}(k+1)\log n = O(\log^2 n)\ . \end{aligned}$$

Donc $T(n) = O(\log^2 n)$.

Question 18. *Montrez que, malheureusement, cette variante peut dans certains cas ne pas trouver*

de minimum local. Exhibez un contre-exemple.

SOLUTION. **[1 pts]** Il faut une colonne avec deux minimum locaux, et choisir en premier le plus grand des deux. Puis partir dans le mauvais sous-rectangle avec un chemin gradient qui, quelque soit la position de départ dans le sous-rectangle, passe par l'autre minimum local (le plus petit des deux).

```
8 8 1 2 3 4 5
8 8 8 8 4 5 6
8 8 8 8 5 6 7
8 8 8 7 6 7 8
8 8 8 8 7 8 9
```

Dans cet exemple, le minimum local de la colonne du mileu $\boxed{2}\ \boxed{8}\ \boxed{8}\ \boxed{7}\ \boxed{8}$ va être 7 (après avoir coupé en deux la colonne). Puis on part dans le sous-rectangle de droite, celui contenant le 6. Le seul minimum local dans ce sous-rectangle est 3, qui malheureusement n'est pas un minimum local du rectangle complet. En fait, tous les minimums locaux sont dans la partie de gauche.

FIN.