

Université de Nice-Sophia Antipolis Licence Informatique - 2e année U.E. Structure de Données

2019-2020

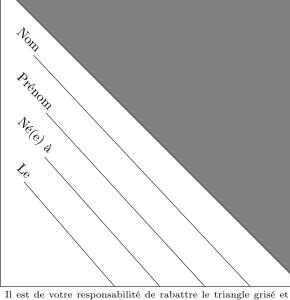
Épreuve de contrôle continu du Jeudi 17 octobre 2019

Durée: 1 heure 15

Tous documents autorisés. Il est interdit d'accéder à internet

Note

Toutes les questions sont indépendantes. Tous les algorithmes devront être écrits en pseudo code. La notation est donnée à titre indicatif.



de le cacheter au moyen de colle, agrafes ou papier adhésif. Si ne vous le faites pas, vous acceptez implicitement que votre copie ne soit pas anonyme.

Exécution d'algorithmes et listes chaînée (8 points) 1

1. Voici l'algorithme du tri par insertion

```
tri_insertion(T[], n) {
pour (i de 2 à n) {
  x \, < \!\! - \, T[\,i\,]
  j\ <\!\!-\ i
   tant que (j > 1 et T[j - 1] > x) {
     T[j] \leftarrow T[j-1]
     j <- j - 1
  T[\;j\;]\;<\!-\;x
```

Exécutez cet algorithme et donnez le nombre de modifications effectuées pour le tableau suivant:

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|----|---|---|---|---|---|----|
| 10 | 2 | 3 | 9 | 8 | 7 | 12 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 10 | 2 | 3 | 9 | 8 | 7 | 12 |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

2. Soit l'algorithme suivant :

```
\begin{array}{c} secret\left(n\right) \; \left\{ \\ x < - \; 1 \\ y < - \; 1 \\ tant \; que \; \left(x < \; n\right) \; \left\{ \\ x < - \; x \; + \; 1 \\ y < - \; y \; + \; 2 \; * \; x \; - \; 1 \\ \right\} \end{array}
```

Exécutez cet algorithme pour $n=4\,$

| Elicource cor | | | |
|---------------|---|--|--|
| X | у | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

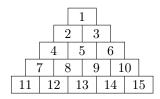
Que fait cet algorithme?

3. Dans le cours, nous avons vu comment parcourir une liste simplement chaînée et y ajouter des éléments. Écrivez l'algorithme ajouteEnFin(elt, L) qui insère l'élément elt après le dernier élément de la liste L.

| | <u> </u> |
|---------------------------|--|
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| 2 Yva | in et Gauvain (4 points) |
| 10 objets d prendre qu | une quête pour le graal, Yvain et Gauvain se retrouvent avec deux sacs contenant chacun e valeurs différentes. Cependant, ayant oublié d'attacher leurs chevaux, ils ne peuvent 'un seul sac à dos. Ils choisissent donc de ne prendre que les dix objets les plus chers |
| | ux sacs à dos. |
| | que: Ces sacs à dos ont la bonne idée d'être rangés en fonction de la valeur des objets: |
| | e position se trouve l'objet le moins cher, et ainsi de suite jusqu'à la dernière position re l'objet le plus cher. |
| | e l'objet le plus cher. Dosant que les sacs à dos sont représentés par des tableaux de $n=10$ éléments, écrivez |
| | e en $O(n)$ qui permet de remplir le sac à dos final qu'ils rapporteront à la table ronde. |
| | valeur égale, on prendra l'objet qui se trouve dans le premier sac considéré. |
| En cas de . | reaction of product robject qui be trouve dams to promier base considerer |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

3 Pyramide (8 points)

En considérant une pyramide comme suit :



On peut la considérer dans la suite, comme un tableau à une dimension (on linéarise la pyramide). La pyramide dessinée ci-dessus serait alors vue comme le tableau :

- Trouvez la fonction qui permet de retrouver la case dans le tableau si on donne le numéro de l'élément dans la ligne de la pyramide : par exemple, le 3e élément de la 5e ligne est la case 13.
 Quel est le code pour retrouver les deux fils d'un élément? Celui pour retrouver ses deux ancêtres?
- 3. Si maintenant on met dans la racine une valeur k, et qu'on remplit la pyramide de la façon suivante : le fils contient la somme des deux pères ou k si un seul père. Écrivez l'algorithme qui permet de remplir de telle façon la pyramide. **Rappel** : La pyramide est représentée par un tableau.

