Informatique tronc commun TP 06 : Algorithmes sur les tableaux.

30 novembre 2016

1. **Lisez attentivement tout l'énoncé avant de commencer.**
2. Commencez la séance en créant un dossier au nom du TP dans le répertoire dédié à l'informatique de votre compte.
3. Après la séance, vous devez rédiger un compte-rendu de TP et l'envoyer au format électronique à votre enseignant.
4. Vous rendrez un compte-rendu pour chaque séance sous forme d'un fichier d'extension `.py`, en respectant exactement les spécifications données plus bas.
5. Ce TP est à faire en binôme, vous ne rendrez donc qu'un compte-rendu pour deux.
6. Ayez toujours un crayon et un papier sous la main. Quand vous réfléchissez à une question, utilisez les !
7. Vous devez être autonome. Ainsi, avant de poser une question à l'enseignant, merci de commencer par :
   - relire l'énoncé du TP (beaucoup de réponses se trouvent dedans) ;
   - relire les passages du cours [1] relatifs à votre problème ;
   - effectuer une recherche dans l'aide disponible sur votre ordinateur (ou sur internet) concernant votre question.

   Il est alors raisonnable d'appeler votre enseignant pour lui demander des explications ou une confirmation !

Le but de ce TP est de travailler sur les tableaux (ou listes Python). Plus précisément, nous allons nous intéresser à la recherche d'un plus long sous-tableau croissant d'un tableau d'entiers.

Instructions de rendu

Attention : suivez précisément ces instructions. Votre fichier portera un nom du type `tpXX_berne_zannad.py`, où `XX` est à remplacer par le numéro du TP et les noms de vos

1. Dans le cas fort improbable où vous ne vous en souviendriez pas.

enseignants par ceux des membres du binôme. Le nom de ce fichier ne devra comporter ni espace, ni accent, ni apostrophe. Dans ce fichier, vous respecterez les consignes suivantes.

— Écrivez d'abord en commentaires (ligne débutant par #), le titre du TP, les noms et prénoms des étudiants du groupe.
— Commencez chaque question par son numéro écrit en commentaires.
— Les questions demandant une réponse écrite seront rédigées en commentaires.
— Les questions demandant une réponse sous forme de fonction ou de script respecterons pointilleusement les noms de variables et de fonctions demandés.

1 Plus long sous-tableau croissant

Étant donné un tableau $t = [t_0, \dots, t_{n-1}]$ d'entiers de longueur n, on appelle sous-tableau croissant de t un tableau $[t_{i_0}, \dots, t_{i_{k-1}}]$ (donc, de longueur k), avec

— $0 \leq i_0 < i_1 < \cdots < i_{k-1} \leq n-1$;
— $t_{i_0} \leq t_{i_1} \leq \cdots \leq t_{i_{k-1}}$.

On considèrera qu'un tableau vide, donc de longueur 0, est croissant.

On s'intéresse au problème de la recherche du plus long sous-tableau croissant dans un tableau d'entiers.

Q1 Quelle est la longueur minimale d'un tel plus long sous-tableau croissant ? Quand est-elle atteinte ?

Q2 Que dire si l'on trouve un plus long sous-tableau croissant de longueur n ?

Q3 Écrire une fonction `est_croissant(t)` renvoyant un booléen indiquant si le tableau d'entiers `t` est croissant.

1.1 Recherche exhaustive

On cherche d'abord à mettre en œuvre une stratégie naïve : on explore tous les sous-tableaux de t (là, vous devriez vous dire que ce n'est pas une bonne idée) !

Pour faire cela, rappelons nous que prendre un sous-tableau de t correspond à prendre une partie de $[\![0, n[\![$. Or, l'ensemble des parties de $[\![0, n[\![$ est en bijection avec $\{0,1\}^{[\![0,n[\![}$. Un sous-tableau v de t est donc caractérisé par une famille $(e_0, \dots, e_{n-1}) \in \{0,1\}^{[\![0,n[\![}$, avec, si $0 \leq i < n$, $e_i = 1$ si et seulement si $t[i]$ est présent dans v. Enfin, on peut voir que pour obtenir toutes les familles de $\{0,1\}^{[\![0,n[\![}$, il suffit d'écrire la décomposition binaire sur n bits de tous les entiers entre 0 et $2^n - 1$.

Exemple 1.1.1. Avec `t = [1,5,3,2]`, de longueur 4, le sous-tableau de `t` décrit par `[0,0,0,0]` est `[]`, celui décrit par `[1,0,0,1]` est `[1,2]` et celui décrit par `[0,1,1,1]` est `[5,3,2]`.

On peut alors penser à implémenter l'idée suivante : on parcourt séquentiellement les entiers entre 0 et $2^n - 1$, on calcul l'écriture binaire de chaque entier, cela définit un sous-tableau de t et il ne reste plus qu'à savoir si ce sous-tableau est croissant. On se rend compte rapidement que cette idée est peu efficace : on va calculer beaucoup de décompositions binaires ! On va plutôt parcourir directement ces décompositions binaires.

Q4 Écrire une fonction `plus_un(e)` qui prend en argument un tableau `e` contenant des 0 et des 1 et qui, si `e` représente l'entier p en binaire, le modifie pour qu'il représente l'entier $p+1$ (modulo 2^n).
Indice : on partira de la fin de `e` et l'on réfléchira à la propagation de la retenue : tant que l'on rencontre des 1*, on les change en* 0 *; dès que l'on rencontre un* 0*, on le change en* 1 *et l'on s'arrête.*

Exemple 1.1.2. À partir de `e=[0,0,0]`, `plus_un(e)` change `e` en `[0,0,1]`, ensuite `plus_un(e)` change `e` en `[0,1,0]` etc.

Q5 Écrire une fonction `longueur(e)` qui prend en argument un tableau `e` contenant des 0 et des 1 et qui renvoie le nombre de 1 dans `e`.

Q6 Écrire une fonction `extrait(t,e)` qui prend en argument un tableau `t` d'entiers et un tableau `e` de 0 et de 1 et qui renvoie le sous-tableau de `t` désigné par `e`.

Q7 Écrire une fonction `stc_exhaustif(t)` donnant la longueur du plus grand sous-tableau croissant de `t`.

1.2 Programmation dynamique

Le programme écrit dans la partie précédente n'est pas très efficace (essayez par exemple de le tester sur un tableau de longueur 100). Nous allons adopter une autre statégie.

Toujours avec $t = [t_0, \dots, t_{n-1}]$ un tableau d'entiers de longueur n, on note $m = [m_0, \dots, m_{n-1}]$ le tableau de longueur n tel que, si $0 \leq i < n$, m_i est la taille du plus grand sous-tableau croissant de t dont le dernier élément est t_i.

Q8 Que vaut m_0 ?

Q9 Soit $0 \leq i < n-1$, supposons que l'on connaisse t et $[m_0, \dots, m_i]$. Comment calculer m_{i+1} ?

Q10 Si l'on connaît m, comment obtenir la longueur de la plus grande sous-suite croissante de t ?

Q11 Écrire une fonction `stc_dynamique(t)` donnant la longueur du plus grand sous-tableau croissant de `t`.

Q12 Quelle est la différence entre les fonctions écrites dans chaque partie, notamment à l'utilisation ?