

Nom :

Devoir 7

Soit une liste  $L$  non vide. On cherche à trouver la valeur maximale de cette liste en utilisant une fonction récursive  $maxi(L)$  :

- Le cas trivial est le cas de la liste de longueur unitaire
- En dehors de ce cas, on compare les deux derniers éléments de la liste.
  - Si  $L[-2] > L[-1]$  on lance un nouvel appel récursif  $maxi(L[0:-1])$
  - Si  $L[-2] < L[-1]$  on lance un nouvel appel récursif  $maxi(L[0:-2] + [L[-1]])$

1) Ecrire le programme associé à  $maxi(L)$  :

Soit la suite définie par : 
$$\begin{cases} u_0 = 1 \\ u_{n+1} = \frac{1}{2}u_n + 1 \end{cases}$$

2) Ecrire une fonction récursive qui reçoit un entier  $n$  et renvoie la valeur de  $u_n$  de la suite.

3) Proposer une fonction récursive  $recherche(L, x)$  qui renvoie un booléen indiquant la présence (True) ou non (False) de  $x$  dans  $L$  (liste triée de manière croissante) par dichotomie en utilisant des slices. Le cas trivial étant celui d'une liste vide

Nom :

Devoir 7

Soit une liste  $L$  non vide. On cherche à trouver la valeur maximale de cette liste en utilisant une fonction récursive  $maxi(L)$  :

- Le cas trivial est le cas de la liste de longueur unitaire
- En dehors de ce cas, on compare les deux derniers éléments de la liste.
  - Si  $L[-2] > L[-1]$  on lance un nouvel appel récursif  $maxi(L[0:-1])$
  - Si  $L[-2] < L[-1]$  on lance un nouvel appel récursif  $maxi(L[0:-2] + [L[-1]])$

1) Ecrire le programme associé à  $maxi(L)$  :

Soit la suite définie par : 
$$\begin{cases} u_0 = 1 \\ u_{n+1} = \frac{1}{2}u_n + 1 \end{cases}$$

2) Ecrire une fonction récursive qui reçoit un entier  $n$  et renvoie la valeur de  $u_n$  de la suite.

4) Proposer une fonction récursive  $recherche(L, x)$  qui renvoie un booléen indiquant la présence (True) ou non (False) de  $x$  dans  $L$  (liste triée de manière croissante) par dichotomie en utilisant des slices. Le cas trivial étant celui d'une liste vide

Nom :

Devoir 7

Nom :

Devoir 7