

Nom :

Devoir 6

Soit une liste L d'entiers dont on cherche la plus petite valeur. Cette liste est non vide.	
1) Proposer un algorithme itératif permettant de trouver cette valeur.	
2) Proposer un algorithme récursif permettant de trouver ce minimum. - le cas trivial est celui pour lequel la liste est de longueur unitaire - à chaque appel on compare les deux premiers éléments x1 et x2 de la liste - le nouvel appel se fait avec la liste initiale dépourvue de la valeur la plus grande entre x1 et x2	
3) Commenter la complexité spatiale du programme précédent.	
4) Pour améliorer le programme précédent, proposer un programme récursif que ne modifie pas la liste L mise en argument. Le principe reste le même sauf que cette fonction utilisera un argument n supplémentaire. Cet argument permettra de désigner les éléments L[n] et L[n+1] qui sera tour à tour comparer pour trouver le minimum.	
5) Commenter la complexité spatiale de ce programme.	

Nom :

Devoir 6

Soit une liste L d'entiers dont on cherche la plus petite valeur. Cette liste est non vide.	
1) Proposer un algorithme itératif permettant de trouver cette valeur.	
2) Proposer un algorithme récursif permettant de trouver ce minimum. - le cas trivial est celui pour lequel la liste est de longueur unitaire - à chaque appel on compare les deux premiers éléments x1 et x2 de la liste - le nouvel appel se fait avec la liste initiale dépourvue de la valeur la plus grande entre x1 et x2	
3) Commenter la complexité spatiale du programme précédent.	
4) Pour améliorer le programme précédent, proposer un programme récursif que ne modifie pas la liste L mise en argument. Le principe reste le même sauf que cette fonction utilisera un argument n supplémentaire. Cet argument permettra de désigner les éléments L[n] et L[n+1] qui sera tour à tour comparer pour trouver le minimum.	
5) Commenter la complexité spatiale de ce programme.	

Nom :

Devoir 6

Nom :

Devoir 6