

Algorithmique

Correction Partiel n° 1 (P1)

INFO-SUP S1 – EPITA

4 janvier 2022 - 9h30

Solution 1 (Colline – 4 points)

Spécifications :

La fonction `hill(L)` détermine si la liste L contenant des entiers naturels est une colline. Si c'est le cas, elle retourne son point culminant (la valeur la plus haute), -1 sinon. Si L est vide la fonction déclenche une exception.

```
1 def hill(L):
2     if len(L) == 0: # L == [] is also authorized
3         raise Exception("Empty list")
4
5     i = 0
6     n = len(L)
7     while i < n-1 and L[i] < L[i+1]:
8         i = i + 1
9
10    top = L[i]
11    i = i + 1
12
13    while i < n-1 and L[i] > L[i+1]:
14        i = i + 1
15
16    if i >= n-1:
17        return top
18    else:
19        return -1
```

Solution 2 (Suppression dans liste triée – 4 points)

Spécifications :

La fonction `delete(L, x)` supprime la valeur `x`, si elle existe, dans la liste `L` strictement décroissante et retourne un booléen indiquant si la suppression a pu être effectuée.

```
1  def delete(L, x):
2      n = len(L)
3      i = 0
4      while i < n and x < L[i]:
5          i += 1
6      if i == n or L[i] != x:
7          return False
8      else:
9          for j in range(i, n-1):
10             L[j] = L[j+1]
11             L.pop()
12             return True
```

Solution 3 (Codage RLE simplifié – 8 points)

Spécifications :

La fonction `decodeRLE` décompresse une liste compressée en RLE.

```
1  def decodeRLE1(L):
2      R = []
3      for i in range(len(L)):
4          (n, val) = L[i]
5          for j in range(n):
6              R.append(val)
7      return R
8
9  def decodeRLE2(L):
10     R = []
11     for (n, val) in L:
12         for _ in range(n):
13             R.append(val)
14     return R
15
16  def decodeRLE3(L):
17     R = []
18     (i, n) = (0, len(L))
19     while i < n:
20         (nb, val) = L[i]
21         while nb > 0:
22             R.append(val)
23             nb -= 1
24         i += 1
25     return R
```

Spécifications :

La fonction `encodeRLE` compresse une liste en utilisant l'algorithme RLE.

```
1     def encodeRLE(L):
2         R = []
3         if L != []:
4             (nb, val) = (1, L[0])
5             for i in range(1, len(L)):
6                 if L[i] == val:
7                     nb += 1
8                 else:
9                     R.append((nb, val))
10                    (nb, val) = (1, L[i])
11            R.append((nb, val))
12    return R
```