Algorithmics
Correction Final Exam #1 (P1)

Undergraduate 1st year S1 – Epita

8 Jan. 2019 - 10 : 00

Solution 1 **(Binary Search: search "path" –** ***3 points*****)**

①	50 - 15 - 48 - 22 - 46 - 42	YES –	
②	48 - 15 - 45 - 22 - **47** - 42		– NO
③	15 - 22 - 45 - 43 - 35 - 42	YES –	
④	22 - 45 - 43 - **15** - 35 - 42		– NO

Solution 2 **(Searching algorithms –** ***2 points*****)**

1. Linear search regardless of element order: 13
2. Linear search taking into account the element order: 9
3. Binary search: $8 = 2 \times 4$

Solution 3 **(See Syracuse –** ***3 points*****)**

Specifications:
The function `Syracuse`(n) builds the list L of all the Syracuse sequence numbers from n if $n \geq 1$. Otherwise, it returns an empty list.

```python
def Syracuse(n):
    if n <= 0:
        return []
    else:
        L = [n]
        while n != 1:
            if n % 2 == 0:
                n = n // 2
            else:
                n = 3 * n + 1
            L.append(n)
        return L
```

Solution 4 (Arithmetic progression – *4 points*)

Specifications:

The function `arithmetic(L)` tests whether the list L has at least two elements and follows an arithmetic progression. In this case, it returns the common difference, otherwise it returns 0.

Principe

If the list has at least two elements, the possible common difference is the two first elements difference. The list is traveled as long as the current element added to the common difference gives the next element. If the travel reaches the last element, then the property is verified.

```python
def arithmetic(L):
    n = len(L)
    if n < 2 or L[1] - L[0] == 0:
        return 0
    else:
        diff = L[1] - L[0]
        i = 1
        while i < n and L[i-1] + diff == L[i]:
            i += 1
        if i == n:
            return diff
        else:
            return 0
```

Solution 5 (Deletion in sorted list – *5 points*)

Specifications:

The function `delete(L, x)` removes the value x, if it exists, from the list L sorted in strictly increasing order and returns a boolean that indicates whether the deletion occurred.

Principe :

- Search:
  Beginning at the first element, the list is traveled as long as the current element does not exceed or reach x.
- Deletion:
  If x has been found (if the list has not been travelled entirely and the element on which the travel stopped is x):
  - All the values after x are shifted from one place to the left.
  - The last cell is "deleted".

```python
def delete(L, x):
    n = len(L)
    i = 0
    while i < n and x > L[i]:
        i += 1
    if i == n or L[i] != x:
        return False
    else:
        for j in range(i, n-1):
            L[j] = L[j+1]
        L.pop()
        return True
```

Solution 6 **(What is it? –** ***3 points*****)**

1. *Result of the following application of* `what`*:*

```
>>> what([1,3,2,8,7,2,5,4,0,6,2,15])
[0, 1, 2, 2, 2, 3, 4, 5, 6, 7, 8, 15]
```

2. We call `what(`L`)` with L a list of natural numbers.

   (a) At the end of the first loop `me` represents the maximum value of L (if L non empty, otherwise it is 0).

   (b) At the end of the third loop `X` represents an histogram of value in L.

   (c) The function returns a coy of L sorted in increasing order.

3. **Bonus:** Complexity: $3n + me$ with n the length of L ($O(max(n, me))$).