

```

from random import randint
from math import ceil

# Tìm phần tử lần chiếm cấp độ 10
def findNum(array):
    store = {}
    for n in array:
        if n in store.keys():
            store[n] += 1
        else:
            store[n] = 1

    result = []
    for key in store.keys():
        if store[key] >= ceil(len(array) / 10):
            result.append(key)

    return result

input_array = [randint(0, 10) for i in range(100)]
print(findNum(input_array))

```

[2, 7, 5, 4, 1]

Ta có 2 vòng lặp for. Vòng lặp for thứ nhất sẽ duyệt qua hết mảng đầu vào. Trong quá trình này ta sẽ lưu dictionary, key là số trong mảng đầu vào, giá trị của key là số lần số đó xuất hiện trong mảng. Vòng lặp for thứ hai sẽ duyệt qua dictionary, key nào có số lần xuất hiện lớn hơn $n/10$ thì ta sẽ lưu vào danh sách kết quả. Cuối cùng ta trả về danh sách kết quả.

Độ phức tạp thời gian của thuật toán là $O(n) + O(k)$ với k là số các phần tử khác nhau trong mảng đầu vào. Trường hợp k lớn nhất, khi đó $k = n$, tức tất cả phần tử trong mảng đầu vào khác nhau, độ phức tạp khi đó sẽ là $O(n) + O(n) = O(2n) = O(n)$. Chứng minh bằng định nghĩa Big O, chọn $c = 2$, $N_0 = 0$.

```

def merge_array(A, B):
    result = []
    n1 = len(A)
    n2 = len(B)
    i = 0
    j = 0

    while i < n1 and j < n2:
        if A[i] < B[j]:
            result.append(A[i])
            i += 1
        elif A[i] > B[j]:
            result.append(B[j])
            j += 1
        else:
            result.append(A[i])
            result.append(B[j])
            i += 1
            j += 1

    while i < n1:
        result.append(A[i])
        i += 1
    while j < n2:
        result.append(B[j])
        j += 1

    return result

```

```

A = [1,3,5,6,7]
B = [3,4,8,9,10]
print(merge_array(A, B))

```

[1, 3, 3, 4, 5, 6, 7, 8, 9, 10]

Merge 2 mảng đã được sắp xếp. Ta tạo một mảng kết quả, sau đó lần lượt lấy ra các số trong 2 mảng đầu vào và so sánh với nhau. 2 biến n1, n2 dùng để lưu độ dài của 2 mảng đầu vào (ở đây cả hai bằng nhau và bằng n). 2 biến i và j dùng là index hiện tại của 2 mảng A, B. Nếu số đầu của A của bé hơn số đầu của B, ta thêm số đầu của A vào kết quả rồi lấy số thứ hai của A so sánh với số B. Cứ thế cho đến khi một trong hai mảng đã được duyệt hết, sau đó ta sẽ thêm các phần tử chưa xét của mảng còn lại vào mảng kết quả.