

BÁO CÁO THỰC HÀNH MÔN PHÂN TÍCH THUẬT TOÁN**Lab 9 - Tuần 13****Đinh Anh Huy - 18110103**

Bài toán. Ứng với số tự nhiên n , ta gọi $f(n)$ là tổng bình phương tất cả các chữ số của n như sau:

Cho trước số tự nhiên N bất kỳ, ta xây dựng dãy số sau đây:

$$a_{n+1} = f(a_n) \quad \text{và} \quad a_0 = N.$$

Viết thuật toán tìm chu kỳ tuần hoàn nhỏ nhất của dãy số đó với giá trị N cho trước.

Ví dụ: nếu $N = 85$, dãy sẽ có giá trị lần lượt như sau:

$$85 \rightarrow 89 \rightarrow 145 \rightarrow 42 \rightarrow 20 \rightarrow 4 \rightarrow 16 \rightarrow 37 \rightarrow 58 \rightarrow 89 \rightarrow 145 \rightarrow \dots$$

Lời giải

Ý tưởng cho thuật toán tìm chu kỳ tuần hoàn nhỏ nhất của dãy số trên như sau: Mỗi lần ta tính được giá trị mới cho dãy, ta sẽ xét xem có phần tử nào trước đó của dãy bằng giá trị hiện tại vừa tính được hay không, nếu có thì ta tìm được chu kỳ tuần hoàn nhỏ nhất là dãy các số từ số có giá trị bằng giá trị hiện tại đến số vừa tìm được ở bước trước đó.

Sử dụng 2 thuật toán phụ sau để giúp thuật toán chính được gọn hơn:

- Thuật toán tìm phần tử có giá trị cho trước có trong mảng (mảng không được sắp xếp). Ở thuật toán này, ta xét từng giá trị trong mảng từ trái sang phải và kiểm tra với giá trị cho trước. Thuật toán này sẽ trả về vị trí của giá trị đầu tiên bằng với giá trị cho trước trong mảng, nếu không có thì trả về -1.

Algorithm 1: Thuật toán tìm vị trí của phần tử có giá trị cho trước trong mảng.

Function `search(arr, value):`

```
    N ← length(arr);  
    for i = 0 → N - 1 do  
        if arr[i] = value then  
            return i  
        end  
    end  
    return -1;
```

- Thuật toán tính tổng bình phương các chữ số của số cho trước: Ở thuật toán này, ta xem từng chữ số là một ký tự và số cho trước là 1 chuỗi, khi đó ta chỉ cần lấy từng

phần tử của chuỗi chuyển sang kiểu *int* và lấy bình phương, sau đó tổng tất cả các giá trị vừa tính lại.

Algorithm 2: Thuật toán tính tổng bình phương các chữ số của một số cho trước bằng Python.

```
Function sum-of-squares-of-digits(value):  
    | return sum(int(c) ** 2 for c in str(value))
```

Thuật toán chính:

Algorithm 3: Thuật toán tìm chu kỳ tuần hoàn nhỏ nhất của dãy số.

```
Function min-cycle-permutation(N):  
    | seq  $\leftarrow$  [N];  
    | while True do  
    |     | current  $\leftarrow$  sum-of-squares-of-digits(seq[-1]);  
    |     | add current to seq;  
    |     | temp_seq  $\leftarrow$  seq[: -1];  
    |     | idx  $\leftarrow$  search(temp_seq, current);  
    |     | if idx  $\neq$  -1 then  
    |     |     | return seq[idx : -1];  
    |     | end  
    |     | else  
    |     |     | continue  
    |     | end  
    | end
```
