

Bài tập thực hành 1

15-06-2021

Viết các chương trình sau bằng đệ quy.

Bài 1. Tính tổng các số trong một danh sách số.

Bài 2. Tính ước chung lớn nhất của hai số a và b . Giả sử $a > b$, nếu a chia hết cho b thì $\text{UCLN}(a, b) = b$. Nếu không thì $\text{UCLN}(a, b) = \text{UCLN}(b, a \% b)$.

Bài 3. Đếm số lần xuất hiện của một ký tự trong một chuỗi bất kỳ. Ví dụ:

- `dem_ky_tu('abcbacaa', 'a')` -> 5.
- `dem_ky_tu('abcbacaa', 'd')` -> 0.

Bài 4. In ra màn hình các chuỗi nhị phân có độ dài 5 mà không có 2 ký tự 1 đứng liền nhau.

Bài 5. Cho chương trình tìm kiếm nhị phân trong python như sau:

```
def tim_nhi_phan(A, k, l, r):  
    """  
    A: danh sách (list) số  
    k: số cần tìm  
    l: vị trí bắt đầu  
    r: vị trí kết thúc  
    Gọi tim_nhi_phan(A, k, 0, len(A) - 1) để tìm kiếm trong toàn danh sách  
    """  
  
    # tìm điểm chia đôi  
    m = (l + r) // 2  
    # giải quyết từng trường hợp  
    if l > r: # không tìm thấy  
        return -1  
    if A[m] == k:  
        return m  
    elif A[m] > k: # giá trị cần tìm nằm bên trái điểm chia  
        return tim_nhi_phan(A, k, l, m - 1)  
    else: # giá trị cần tìm nằm bên phải điểm chia  
        return tim_nhi_phan(A, k, m + 1, r)
```

Hãy thay đổi chương trình trên để thực hiện các thuật toán:

- Tìm kiếm tam phân: tương tự như tìm kiếm nhị phân, nhưng thay vì chia đoạn cần tìm làm hai phần, chia nó thành ba phần.
- Tìm kiếm nội suy: tương tự như tìm kiếm nhị phân, nhưng điểm chia sẽ phụ thuộc vào giá trị của số cần tìm k . Trong thuật toán này, chúng ta sẽ dùng điểm chia là số lớn nhất nhỏ hơn $\frac{k - A_l}{A_r - A_l} \times (r - l)$