

# Bài tập thực hành

## Cấu trúc dữ liệu và thuật toán

### Cây tìm kiếm nhị phân (B)

Định dạng chung cho file INPUT.TXT (phần mô tả cây):

- Dòng đầu tiên chứa số nguyên N là số lượng nút trong cây.
- Dòng thứ hai chứa số nguyên root\_id là ID của nút gốc của cây.
- N dòng tiếp theo, mỗi dòng mô tả một nút theo định dạng: node\_id value left\_child\_id right\_child\_id.
  - node\_id: Số nguyên duy nhất định danh nút (ví dụ từ 0 đến N-1).
  - value: Giá trị số nguyên của nút.
  - left\_child\_id: ID của con trái. Nếu không có con trái, giá trị là -1.
  - right\_child\_id: ID của con phải. Nếu không có con phải, giá trị là -1.

Ví dụ:

```
      0 (10)
     /      \
    1 (5)    2 (15)
   /  \    \
  3 (2) 4 (7) 5 (20)
           /
          6 (17)
```

Cấu trúc file INPUT.TXT cho cây này (7 nút, gốc là 0):

```
7
0
0 10 1 2
1 5 3 4
2 15 -1 5
3 2 -1 -1
4 7 -1 -1
5 20 6 -1
6 17 -1 -1
```

## 1. Bài 1: Kiểm tra cây có phải là BST hợp lệ hay không

Cho một cây nhị phân, hãy kiểm tra xem nó có thỏa mãn tính chất của cây tìm kiếm nhị phân (BST) hay không.

Dữ liệu vào trong file INPUT.TXT chứa mô tả cây nhị phân.

Kết quả ghi ra file OUTPUT.TXT là "YES" nếu cây là BST hợp lệ, ngược lại là "NO".

Ví dụ:

INPUT.TXT

```
7
0
0 10 1 2
1 5 3 4
2 15 -1 5
3 2 -1 -1
4 7 -1 -1
5 20 6 -1
6 17 -1 -1
```

OUTPUT.TXT

YES

## 2. Bài 2: Tìm phần tử kế tiếp (Successor) và phần tử trước đó (Predecessor) của một nút trong BST

Cho một cây tìm kiếm nhị phân (BST) và một giá trị nút X, hãy tìm phần tử kế tiếp (successor: giá trị lớn hơn nhưng gần nhất) và phần tử trước đó (predecessor: giá trị nhỏ hơn nhưng gần nhất) của X.

Dữ liệu vào trong file INPUT.TXT chứa mô tả cây nhị phân. Dòng cuối cùng chứa giá trị X.

Kết quả ghi ra file OUTPUT.TXT gồm 2 số cách nhau bởi khoảng trắng: predecessor và successor. Nếu không tồn tại, ghi -1.

Ví dụ:

INPUT.TXT

```
7
0
0 10 1 2
```

```
1 5 3 4
2 15 -1 5
3 2 -1 -1
4 7 -1 -1
5 20 6 -1
6 17 -1 -1
10
```

```
OUTPUT.TXT
7 15
```

### 3. Bài 3: Đếm số nút trong cây BST nằm trong khoảng giá trị cho trước

Viết chương trình đếm số lượng nút trong cây tìm kiếm nhị phân (BST) có giá trị nằm trong đoạn [low, high].

Dữ liệu vào trong file INPUT.TXT chứa mô tả cây nhị phân. Dòng cuối cùng chứa hai số nguyên low và high.

Kết quả ghi ra file OUTPUT.TXT là số lượng nút thỏa mãn điều kiện.

Ví dụ:

```
INPUT.TXT
7
0
0 10 1 2
1 5 3 4
2 15 -1 5
3 2 -1 -1
4 7 -1 -1
5 20 6 -1
6 17 -1 -1
5 15
```

```
OUTPUT.TXT
4
```

### 4. Bài 4: Tìm phần tử nhỏ thứ K (K-th Smallest Element)

Cho một cây tìm kiếm nhị phân (BST) và một số nguyên dương K, hãy tìm giá trị của phần tử nhỏ thứ K trong cây.

Dữ liệu vào trong file INPUT.TXT chứa mô tả cây nhị phân. Dòng cuối cùng chứa số nguyên K.

Kết quả ghi ra file OUTPUT.TXT là giá trị của phần tử nhỏ thứ K. Nếu không tồn tại, ghi -1.

Ví dụ:

INPUT.TXT

```
7
0
0 10 1 2
1 5 3 4
2 15 -1 5
3 2 -1 -1
4 7 -1 -1
5 20 6 -1
6 17 -1 -1
3
```

OUTPUT.TXT

```
7
```

## 5. Bài 5: Tìm Tổ tiên chung gần nhất (Lowest Common Ancestor - LCA)

Cho một cây tìm kiếm nhị phân (BST) và hai giá trị u và v, hãy tìm tổ tiên chung gần nhất (LCA) của hai giá trị này.

Dữ liệu vào trong file INPUT.TXT chứa mô tả cây nhị phân. Dòng cuối cùng chứa hai số nguyên u và v.

Kết quả ghi ra file OUTPUT.TXT là giá trị của nút tổ tiên chung gần nhất. Nếu không tìm thấy, ghi -1.

Ví dụ:

INPUT.TXT

```
7
0
0 10 1 2
1 5 3 4
2 15 -1 5
3 2 -1 -1
4 7 -1 -1
```

5 20 6 -1  
6 17 -1 -1  
2 7

OUTPUT.TXT  
5