

## Bài tập và Thực hành Chương 5

### CÂY

#### Buổi thực hành 7, 8, 9

**Bài 1.** Viết phương thức **cay.SoNut()** như sau:

- **cay** là một cây nhị phân, mỗi nút có ba vùng: vùng **Info** chứa một số nguyên, vùng **Left** chỉ đến nút con bên trái và vùng **Right** chỉ đến nút con bên phải.
- Phương thức trả về số nút của cây.
- **Gợi ý:** số nút của cây gốc  $p$  bằng số nút của cây con bên trái cộng số nút của cây con bên phải cộng 1.

**Bài 2.** Viết phương thức **cay.ChieuCao()** như sau:

- **cay** là một cây nhị phân, mỗi nút có ba vùng: vùng **Info** chứa một số nguyên, vùng **Left** chỉ đến nút con bên trái và vùng **Right** chỉ đến nút con bên phải.
- Phương thức trả về chiều cao của cây.
- **Gợi ý:** chiều cao của cây gốc  $p$  bằng chiều cao lớn nhất của cây con bên trái và của cây con bên phải rồi cộng 1.

**Bài 3.** Viết phương thức **cay.SoNutLa()** như sau:

- **cay** là một cây nhị phân, mỗi nút có ba vùng: vùng **Info** chứa một số nguyên, vùng **Left** chỉ đến nút con bên trái và vùng **Right** chỉ đến nút con bên phải.
- **Nút lá** là một nút không có hai cây con bên trái và bên phải và không là nút gốc.
- Phương thức trả về số nút lá của cây.

**Bài 4.** Viết phương thức **cay.SoNutTrungGian()** như sau:

- **cay** là một cây nhị phân, mỗi nút có ba vùng: vùng **Info** chứa một số nguyên, vùng **Left** chỉ đến nút con bên trái và vùng **Right** chỉ đến nút con bên phải.
- **Nút trung gian** là một nút chỉ có một cây con bên trái hoặc bên phải và không là nút gốc.
- Phương thức trả về số nút trung gian của cây.

**Bài 5.** Viết phương thức **cay.KiemTraBST()** như sau:

- **cay** là một cây nhị phân, mỗi nút có ba vùng: vùng **Info** chứa một số nguyên, vùng **Left** chỉ đến nút con bên trái và vùng **Right** chỉ đến nút con bên phải.
- Phương thức trả về **true** nếu **cay** là cây **BST**, ngược lại trả về **false**.
- **Lưu ý:** tại mỗi nút, không chỉ so sánh khóa của một nút với các khóa của hai nút con của nút này.

**Bài 6.** Viết phương thức **cay.KiemTraAVL()** như sau:

- **cay** là một cây nhị phân, mỗi nút có ba vùng: vùng **Info** chứa một số nguyên, vùng **Left** chỉ đến nút con bên trái và vùng **Right** chỉ đến nút con bên phải.
- Phương thức trả về **true** nếu **cay** là cây **AVL**, ngược lại trả về **false**.

**Bài 7.** Viết phương thức **cay.Chep()** như sau:

- **cay** là một cây nhị phân, mỗi nút có ba vùng: vùng **Info** chứa một số nguyên, vùng **Left** chỉ đến nút con bên trái và vùng **Right** chỉ đến nút con bên phải.
- Phương thức trả về một cây mới giống hệt hoàn toàn (sao chép) với **cay**.

**Bài 8.** Viết phương thức **cay1.SoSanh(cay2)** như sau:

- **Cay1, cay2** là các cây nhị phân, mỗi nút có ba vùng: vùng **Info** chứa một số nguyên, vùng **Left** chỉ đến nút con bên trái và vùng **Right** chỉ đến nút con bên phải.

- Phương thức trả về *true* nếu hai cây *cay1* và *cay2* giống hệt hoàn toàn, ngược lại trả về *false*.

**Bài 9.** Viết phương thức **cay1.CayCon(cay2)** như sau:

- *cay1*, *cay2* là các cây nhị phân, mỗi nút có ba vùng: vùng *Info* chứa một số nguyên, vùng *Left* chỉ đến nút con bên trái và vùng *Right* chỉ đến nút con bên phải.
- Phương thức trả về *true* nếu *cay2* là cây con của *cay1*, ngược lại trả về *false*.

## Bài tập

**Bài 10.** Viết phương thức **cay.CanBangHoanToan()** như sau:

- **cay** là một cây nhị phân, mỗi nút có ba vùng: vùng **Info** chứa một số nguyên, vùng **Left** chỉ đến nút con bên trái và vùng **Right** chỉ đến nút con bên phải.
- Một cây được gọi là cân bằng hoàn toàn nếu tại mọi nút của cây thì số nút của cây con bên trái và số nút của cây con bên phải hơn kém nhau nhiều nhất là 1.
- Phương thức trả về **true** nếu **cay** là cây cân bằng hoàn toàn, ngược lại trả về **false**.

**Bài 11.** Viết phương thức **cay.BSTTuanTu()** như sau:

- **cay** là một cây nhị phân, mỗi nút có ba vùng: vùng **Info** chứa một số nguyên, vùng **Left** chỉ đến nút con bên trái và vùng **Right** chỉ đến nút con bên phải.
- Nếu cây **BST** có mọi nút chỉ có nút con bên trái hoặc mọi nút chỉ có nút con bên phải thì việc tìm kiếm trên cây này là tìm kiếm tuần tự.
- Phương thức trả về **true** nếu **cay** là cây **BST** nêu trên, ngược lại trả về **false**.