**CÂU HỎI**

**Câu 1:** Hãy trình bày các vấn đề sau: định nghĩa và đặc điểm của cây nhị phân tìm kiếm; các thao tác thực hiện tốt trong kiểu này; hạn chế các kiểu CTDL này?

Cây nhị phân tìm kiếm là cây nhị phân mà giá trị của phần tử bên trái của một node có giá trị nhỏ hơn giá trị node, giá trị của các phần tử bên phải của một node thì lớn hơn giá trị của node đó. Dùng để quản lý một tập các phần từ có số lượng lớn, được cấp phát rời rạc trong bộ nhớ.

Các thao tác thực hiện tốt trong kiểu này là:

* Tìm kiếm phần tử nhanh.
* Thêm phần tử.
* Xóa phần tử.

Hạn chế:

* Việc thêm, xóa node trong cây khá phức tạp, do phải thực hiện nhiều phép so sánh.
* Không có giá trị trùng nhau trong cây nhị phân tìm kiếm.

**Câu 2:** Hãy so sánh cây nhị phân tìm kiếm và các CTDL cơ bản: danh sách đặc, danh sách liên kết, danh sách hạn chế.

So sánh:

|  |  |  |  |
| --- | --- | --- | --- |
| Danh sách đặc | là một danh sách mà các phần tử trong danh sách có cùng kiểu dữ liệu, và được cấp phát liên tục trong bộ nhớ. | #define MAX 100  int a[MAX]  int n | Có thể truy xuất ngẫu nhiên các phần tử trong mảng. |
| Danh sách liên kết | là danh sách mà các phần tử được cấp phát rời rạc nhau, cố định trong bộ nhớ. Mỗi thành phần trong danh sách gồm có 2 thành phần: vùng thông tin và vùng liên kết. | struct node  int info  node \*link  node \*first, \*last | Truy xuất tuần tự theo chiều first. |
| Danh sách hạn chế | là danh sách có thứ tự, khi hoạt động truy xuất, phải luôn tuân theo quy tắc đã đề ra, không thể truy xuất phần tử ngẫu nhiên trong danh sách. | Stack và Queue | Truy xuất theo quy tắt FILO và FIFO. |
| Cây nhị phân tìm kiếm | là cây nhị phân mà giá trị của phần tử bên trái của một node có giá trị nhỏ hơn giá trị node, giá trị của các phần tử bên phải của một node thì lớn hơn giá trị của node đó. | struct node  node \*left  node \*right | Truy xuất theo LNR, LRN, NLR. |