**Câu 1:**

* Định nghĩa và đặc điểm của cây nhị phân tìm kiếm: cây nhị phân tìm kiếm là cây nhị phân mà giá trị (khóa) của phần tử bên trái của của một node có giá trị nhỏ hơn giá trị (khóa) của node,giá trị (khóa) của các phần tử bên phải của một node thì lớn hơn giá trị (khóa) của node đó.
* Thao tác thực hiện tốt trong kiểu dữ liệu này là *tìm một node trong cây nhị phân tìm kiếm***.** Vì nó có khả năng tìm kiếm nhanh, do tính chất , giá trị của một node sẽ lớn hơn các giá trị bên nhánh con bên trái và nhỏ hơn giá trị nhánh con bên phải.
* Hạn chế của là*thêm/xóa node trong cây* vì nó khá phức tạp do thực hiện nhiều phép so sánh và các giá trị trong cây nhị phân không được trùng nhau.

**Câu 2:**

|  |  |  |  |
| --- | --- | --- | --- |
| 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 LIFO (Stack) và FIFO(Queue). |
| 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. |