**CHƯƠNG 4**

**1)**

**Cây nhị phân tìm kiếm** là cây nhị phân mà giá trị của phần tử bên trái của 1 node có giá trị nhỏ hơn giá trị của node , giá trị của phần tử bên phải của 1 node có giá trị lớn hơn giá trị của node đó.

+ Các thao tác thực hiện tốt: chèn và xóa và tìm kiếm

+ Hạn chế: trong trường hợp xấu nhất, cây có thể biến thành 1 DSLK (khi mà mỗi nút đều chỉ có 1 con trừ node lá). Lúc đó các thao tác trên sẽ có độ phức tạp từ log2(n) lên đến O(n). Vì vậy cần có cải tiến cấu trúc của cây nhị phân tìm kiếm để đạt đc chi phí thấp.

2)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Khái niệm | Truy Xuất | Cấu trúc | Thời gian tìm kiếm |
| Cây nhị phân tìm kiếm | Là cây nhị phân mà giá trị bên trái nhỏ hơn node(khóa), giá trị bên phải lớn hơn node(khóa). | Theo các thứ tự LNR,LRN,NLR,… | struct node  {  node \*left  node \*right  } | NHANH |
| Danh sách đặc | Là Danh sách cùng kiểu dữ liệu và được cấp phát liên tục trong bộ nhớ | Truy xuất ngẫu nhiên bất cứ phần tử trong mảng. | #define MAX100  int a[MAX]  int n | CHẬM |
| Danh sách liên kết | Là danh sách có các phần tử được cấp phát rời rạc nhau , cố định trong bộ nhớ. Mỗi thành phần trong danh sách gồm 2 phần là vùng liên kết và vùng thông tin. | Theo thứ tự chiều first | Struct node  {  int info  node\*link  node\*first  \*last  } | CHẬM |
| Danh sách hạn chế | Là danh sách có thứ tự. Hoạt động truy xuất phải tuân theo quy tắc đề ra không được truy xuất phần tử ngẫu nhiên. | Theo quy tắc LIFO, FIFO | Ngăn xếp và hàng đợi | CHẬM |