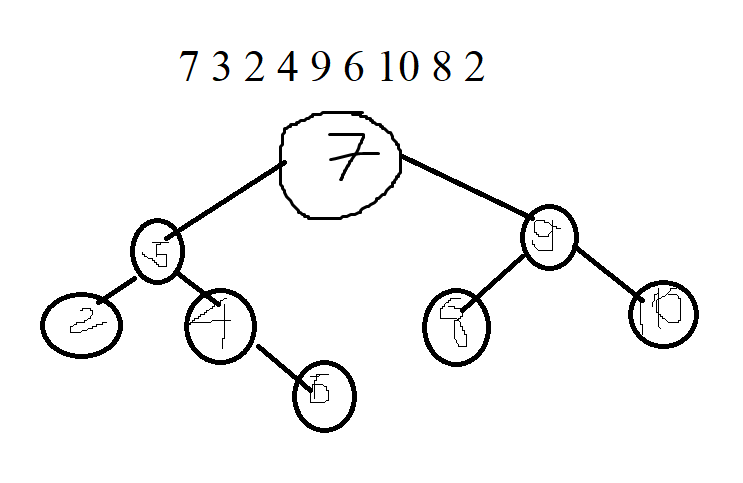
Binary tree

Thuật toán :

Prim

Kruskal  
dijsktra

1. Sử dụng mảng : 

* Phần tử thứ I có 2 con 2i và 2i +1
* Vd : 7 có 2 con 3,9

3 có 2 và 4

9 có 8 và 10

Brr : 1 2 3 4 5 6 7 8 9 10 11

T :

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 3 | 9 | 2 | 4 | 8 | 10 |  |  |  | 6 |

1 2 3 4 5 6 7 8 9 10 11

Thao tác insert, delete, seach

* Init : n = 0;
* Brr// tạo mảng b để xem xem ở ô đó có phần tử hay chưa (0 là chưa có giá trị, 1 là đã có)
* Insert (int x, int \*n, int &pos)

Tại vị trí brr[pos] == 0 gán T[pos] = x , b[pos] = 1; nếu (\*n < pos ) thì \*n = pos; return;

Nếu T[pos] == x; return;

Nếu T[pos] > x thì inset(x,2\*pos, n) ngược lại insert (x,2\*pos+1, n)

* Delete( tree \*t)

+ Nếu t->left != null

Tìm node phải cùng của cây con bên trái t1 = t-> left

While (t1 -> right != null) t1 = t1 -> right

t->data = t1->data

delete(t1)

+ nếu tồn tại cây con bên trái t->right != null

t1 = t-> right

while t1 -> left != null

t1 = t1 ->left

t->data = t1 ->data

delete(t1) // xóa đệ quy

return;

+ th node là lá:

Them con trỏ temp = t

Cho t = null

Free(temp)