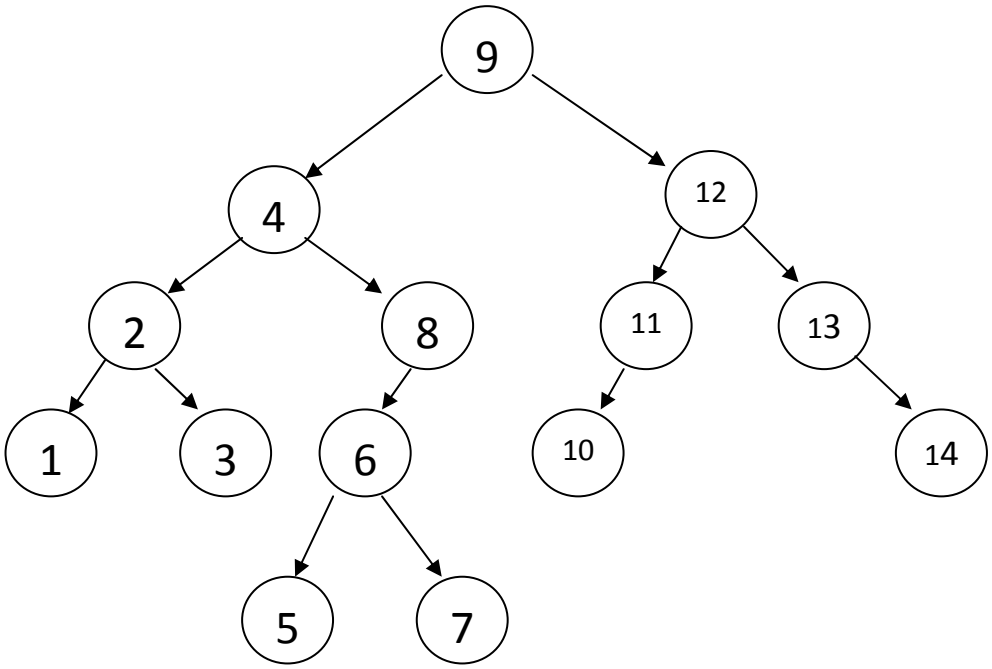
COSTRUIRE UN ALBERO BINARIO DI RICERCA (BST)

Esempio

Sequenza di input:

9 4 12 2 8 11 13 1 3 6 10 14 5 7



```
tree insertBinOrd(element e, tree t){
tree l = t;
if (empty(t))
 return consTree(e,emptyTree(),emptyTree());
while (!empty(t)) {
    if (e <= root(t)){
        if (empty(left(t))){
            t->left =
          consTree(e,emptyTree(),emptyTree());
            t = left(t);}
            t = left(t);
        } else {
            if (empty(right(t))){
                t->right =
         consTree(e,emptyTree(),emptyTree());
                t = right(t);}
            t = right(t);
        }
    }
    return l;
```

```
}
```

COSTRUIRE UN ALBERO BINARIO DI RICERCA (BST)

Visita in PRE-ORDINE:

9 4 2 1 3 8 6 5 7 12 11 10 13 14

Visita in IN-ORDINE:

1 2 3 4 5 6 7 8 9 10 11 12 13 14

Visita in POST-ORDINE:

1 3 2 5 7 6 8 4 10 11 14 13 12 9

```
main () {
tree t;
FILE *f;
element el;
f = fopen("valori.txt", "r");
t = emptyTree();
while (fscanf(f, "%d", &el)>0)
       t = insertBinOrd(el, t);
preOrder(t);
inOrder(t);
postOrder(t);
}
```

CANCELLAZIONE IN UN ALBERO BST

Partendo dalla radice **t** cerco il nodo da eliminare, ricordando il padre di quel nodo

```
while (root(t) != e && !empty(t)) {
    if (e < root(t)){
        pl = t; pr = emptyTree();t = left(t);
    } else {
        pl = emptyTree(); pr = t;  t = right(t);
    }
}
```

pl (pr) indica che il padre del nodo da eliminare lo ha come figlio sinistro (destro), l'altro padre è emptyTree().

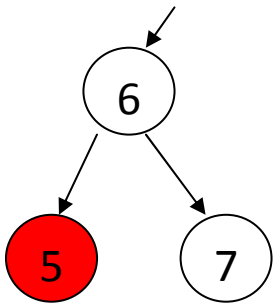
Trovato il nodo da eliminare abbiamo 3 casi possibili:

1. Il nodo da eliminare è foglia
2. Il nodo da eliminare ha un solo figlio
3. Il nodo da eliminare ha due figli

CANCELLAZIONE IN UN ALBERO BST

1. Il nodo da eliminare è foglia: occorre solo eliminare il nodo

Ad esempio voglio eliminare il nodo 5

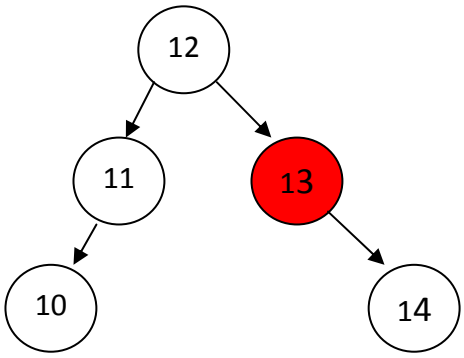



```
if (empty(left(t)) && empty(right(t))){
    if(!empty(pl))
        pl->left = emptyTree();
    else
        pr->right = emptyTree();
```

CANCELLAZIONE IN UN ALBERO BST

2. Il nodo da eliminare ha un solo figlio: occorre sostituire (l'unico) figlio asl nodo da eliminare

Ad esempio voglio eliminare il nodo 13

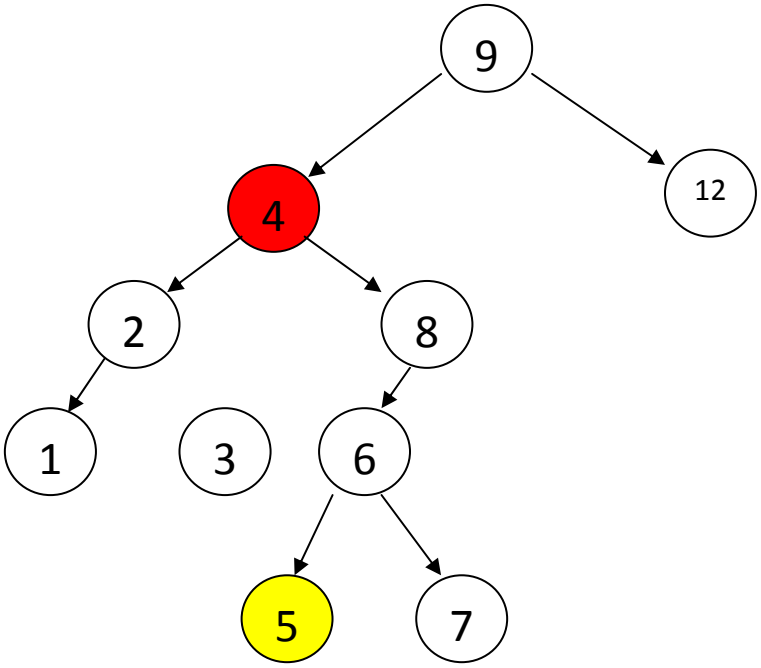



```
if (empty(left(t)) && !empty(right(t))){
    next = right(t);
} else {
   if (!empty(left(t)) && empty(right(t))){
    next = left(t);
}
if(!empty(pl))
    pl->left = next;
else
    pr->right = next;
```

CANCELLAZIONE IN UN ALBERO BST

3. Il nodo da eliminare ha due figli: occorre trovare il nodo minore tra i maggiori del nodo da eliminare (chiamato successore) e sostituirlo con quello eliminato

Ad esempio voglio eliminare il nodo 4



Il succ(t) è il nodo più a sinistra del sottoalbero di destra di t.

Trovato succ(t) occorre sostituirlo a t e poi eliminarlo.

CANCELLAZIONE IN UN ALBERO BST

```
if (!empty(left(t)) && !empty(right(t))){
    pr = t;
    pl = emptyTree();
    next = right(t);
    if (!empty(next))
        while(!empty(left(next))){
            pr = emptyTree(); pl = next;
            next= left(next);}
    t->value = root(next);
    next = emptyTree();

if(!empty(pl))
    pl->left = next;
else
    pr->right = next;
```