In [1]:

```
import collections

class Tree:
    def __init__(self, val, n):
        self.val, self.n = val, n
        if n>1:
            self.lft, self.rht = Tree(val-1,n-1), Tree(val+1,n-1)
        else:
            self.lft, self.rht = None, None

def breadthfirst(tree):
    queue = collections.deque([tree])
    while(queue):
        node = queue.popleft()
        print node.val
        if (node.n>1):
            queue.extend([node.lft,node.rht])

def depthfirst(tree):
    lst = [[tree,0]]
    while lst:
        if lst[-1][1]==0:
            lst[-1][1] = 1
            if lst[-1][0].n>1:
                lst.append([lst[-1][0].lft,0])
        elif lst[-1][1]==1:
            lst[-1][1] = 2
            if lst[-1][0].n>1:
                lst.append([lst[-1][0].rht,0])
        else:
            print lst.pop()[0].val

tree = Tree(10,4)

print "Breadth First Search Results:"
breadthfirst(tree)
print "Depth First Search Results:"
depthfirst(tree)
```

```
Breadth First Search Results:
10
9
11
8
10
10
12
7
9
9
11
9
11
11
13
Depth First Search Results:
7
9
8
9
11
10
9
9
11
10
11
13
12
11
10
```