Tower of Honoi Using recursion

In [1]:
```
import timeit
import pandas as pd
```

In [2]:
```
def toh(n,src,dest,helper):
    global count
    if n==1:
        print('Move disk 1 from',src,'to',dest)
        count = count+1
        return
    toh(n-1,src,helper,dest)
    print('Move disk',n, 'from',src,'to',dest)
    count = count+1
    toh(n-1,helper,dest,src)
```

In [3]:
```
time = []
moves = []
```

In [4]:
```
print('----Tower of Honoi----')
for i in range(1,11):
    print("For n=",i)
    count=0
    print('Steps:')
    start_time = timeit.default_timer()
    toh(i,'A','B','C')
    end_time = timeit.default_timer()
    print('Required time:',end_time-start_time)
    print("Number of moves:",count)
    moves.append(count)
    time.append(end_time-start_time)

df = pd.DataFrame(list(zip(moves,time)),columns =['Moves', 'Required Time'])
df.insert(0, 'N', range(1, 1 + len(df)))
df = df.set_index('N')
```

```
----Tower of Honoi----
For n= 1
Steps:
Move disk 1 from A to B
Required time: 0.0006528999999995122
Number of moves: 1
For n= 2
Steps:
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Required time: 0.0005825000000001523
Number of moves: 3
For n= 3
Steps:
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 3 from A to B
```

In [5]:
```
print("====Conclusion===")
df
```

```
====Conclusion===
```

Out[5]:

N	Moves	Required Time
1	1	0.000653
2	3	0.000583
3	7	0.004966
4	15	0.010905
5	31	0.025334
6	63	0.080255
7	127	0.235577
8	255	0.262539
9	511	0.392409
10	1023	0.985207