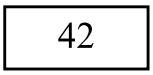
Insertions in B-Trees

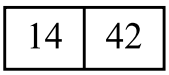
Create a B-tree of degree (order) = 3 Method used: insertions with transformations (split only) in going down

In [1]:
```
from btrees_classics import insert
from algopy.btree import BTree, display2
BTree.degree = 3
```

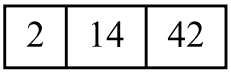
In [2]:
```
B = None
B = insert(B, 42)
display2(B)
```



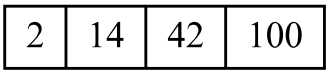
In [3]:
```
B = insert(B, 14)
display2(B)
```



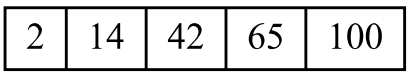
In [4]:
```
B = insert(B, 2)
display2(B)
```



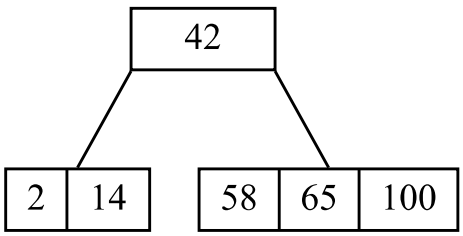
In [5]:
```
B = insert(B, 100)
display2(B)
```



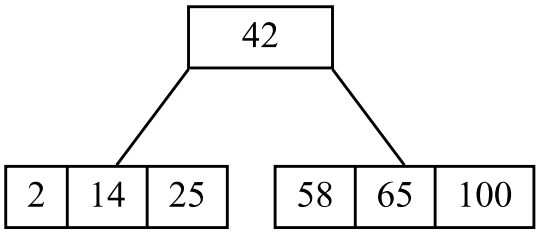
In [6]:
```
B = insert(B, 65)
display2(B)
```



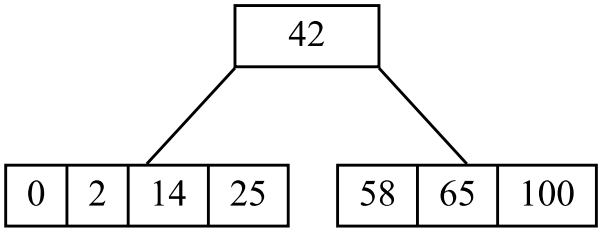
In [7]:
```
B = insert(B, 58) # root is split (a new root -> +1 to height)
display2(B)
```



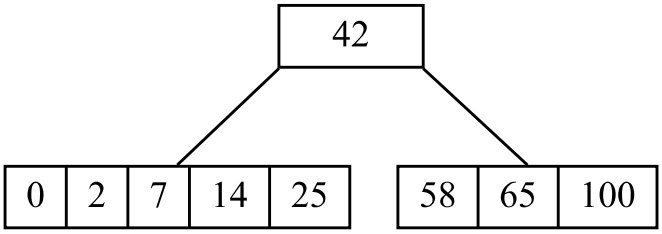
In [8]:
```
B = insert(B, 25)
display2(B)
```



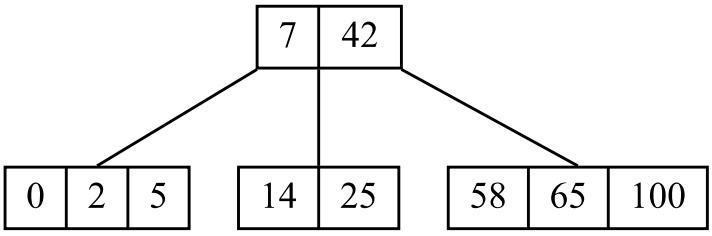
In [9]:
```
B = insert(B, 0)
display2(B)
```



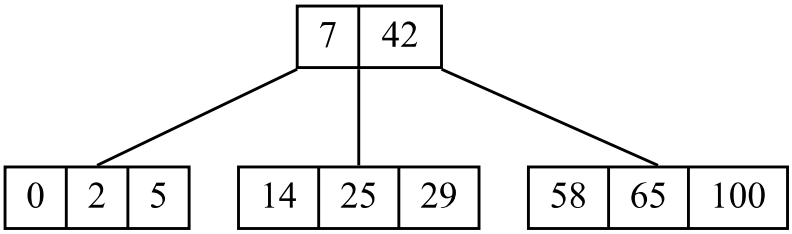
In [10]:
```
B = insert(B, 7)
display2(B)
```



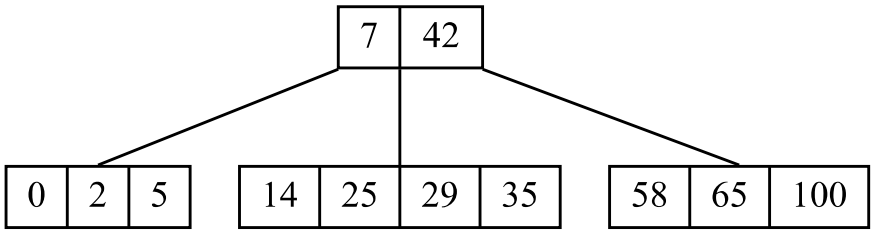
In [11]:
```
B = insert(B, 5) # child 0 was full -> split
display2(B)
```



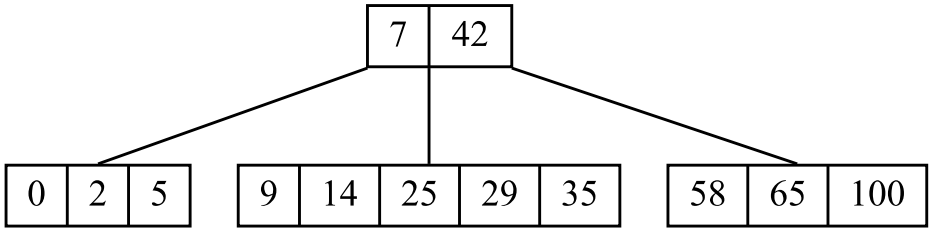
In [12]:
```
B = insert(B, 29)
display2(B)
```



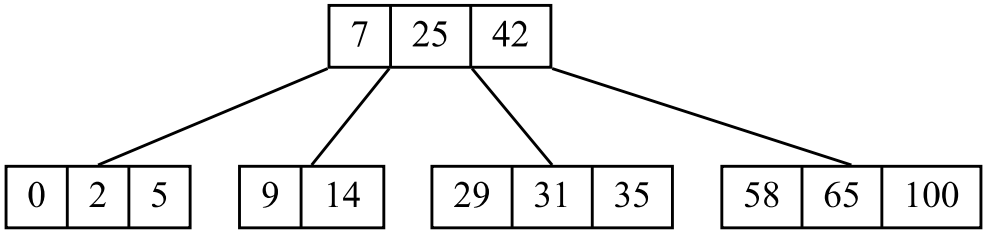
In [13]:
```
B = insert(B, 35)
display2(B)
```



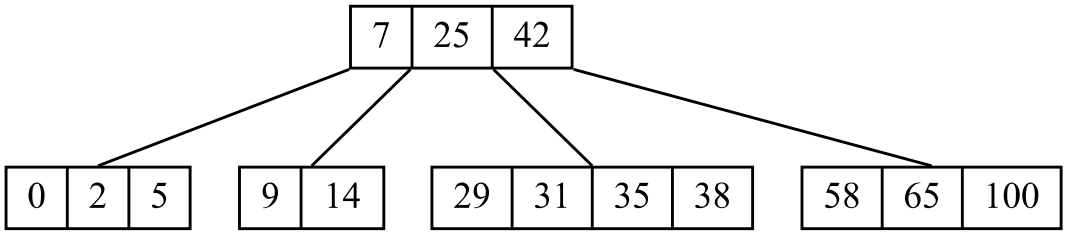
In [14]:
```
B = insert(B, 9)
display2(B)
```



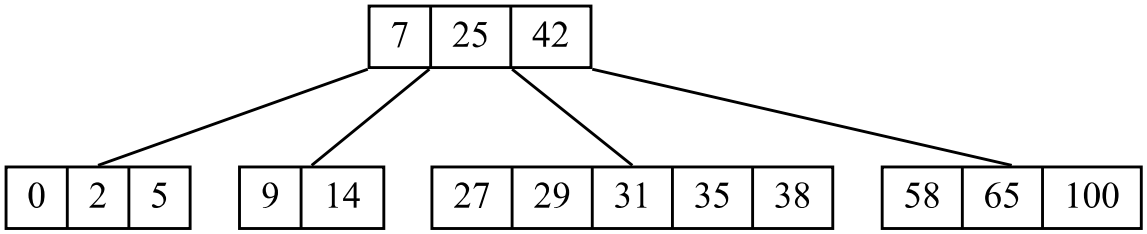
In [15]:
```
B = insert(B, 31) # child 1 was full -> split ->  insertion in child 2 (31 > 25)!
display2(B)
```



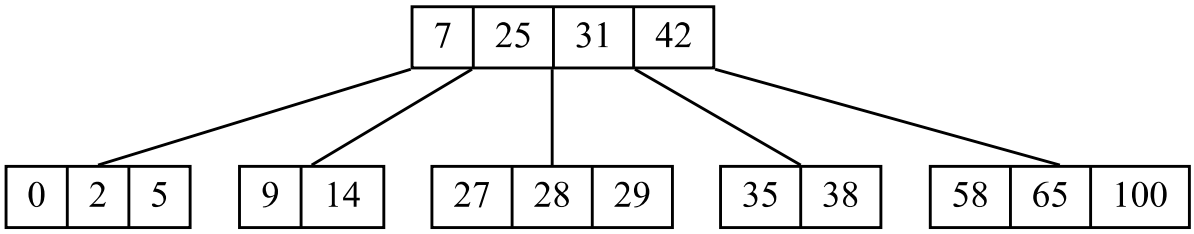
In [16]:
```
B = insert(B, 38)
display2(B)
```



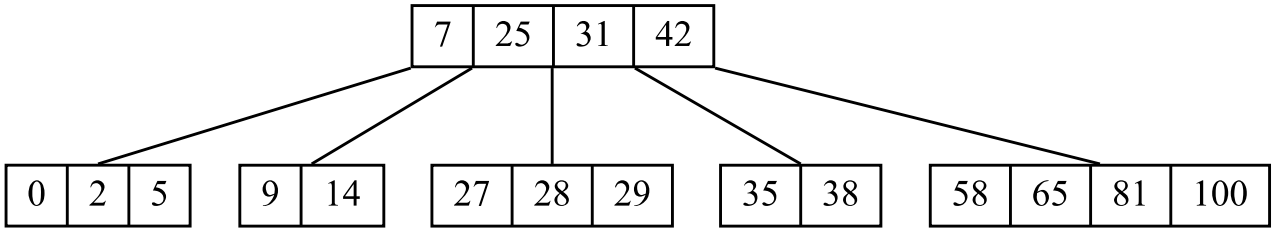
In [17]:
```
B = insert(B, 27)
display2(B)
```



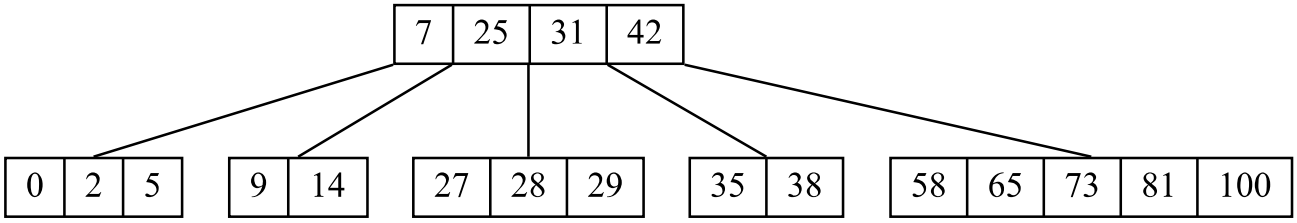
In [18]:
```
B = insert(B, 28) # child 2 was full -> split
display2(B)
```



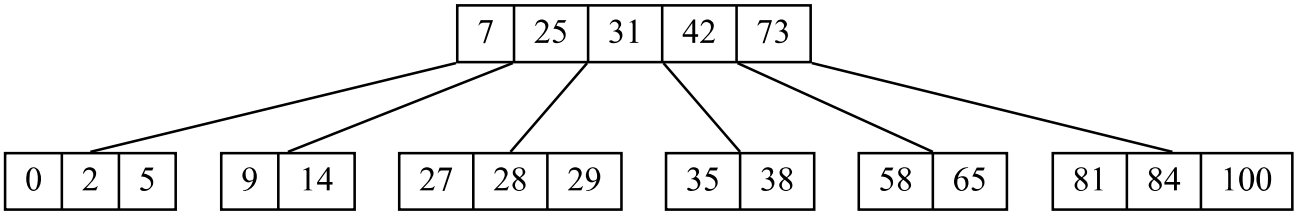
In [19]:
```
B = insert(B, 81)
display2(B)
```



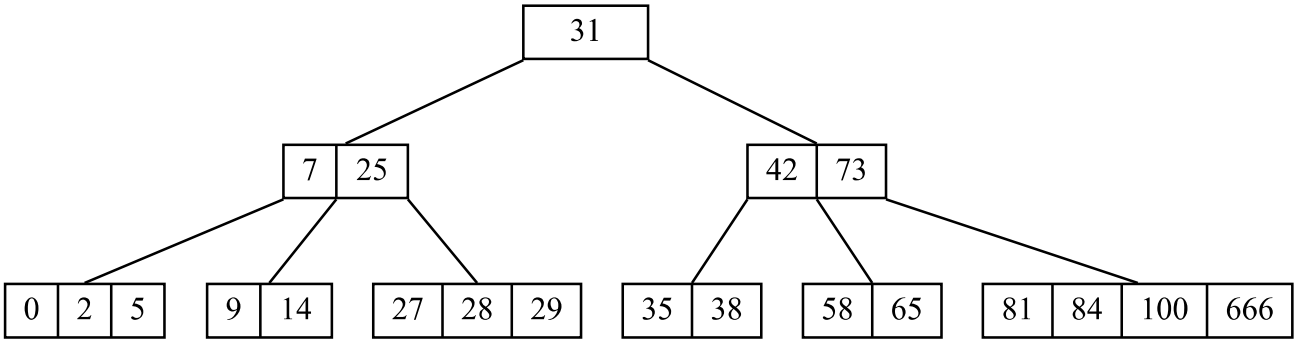
In [20]:
```
B = insert(B, 73)
display2(B)
```



In [21]:
```
B = insert(B, 84) # child 4 was full -> split ->  insertion in child 5 (84 > 73)!
display2(B)
```



In [22]:
```
B = insert(B, 666)   # root is split, even if there was enough place in last child fo
r 666!
display2(B)
```



In [ ]: