Assignment #4

6 points

1. Tree traversals.

Give the sequence of letters for each traversal of this binary tree:

```
       q
     /   \
    e     r
   / \   / \
  c  d  n   s
 /         /
a         w
```

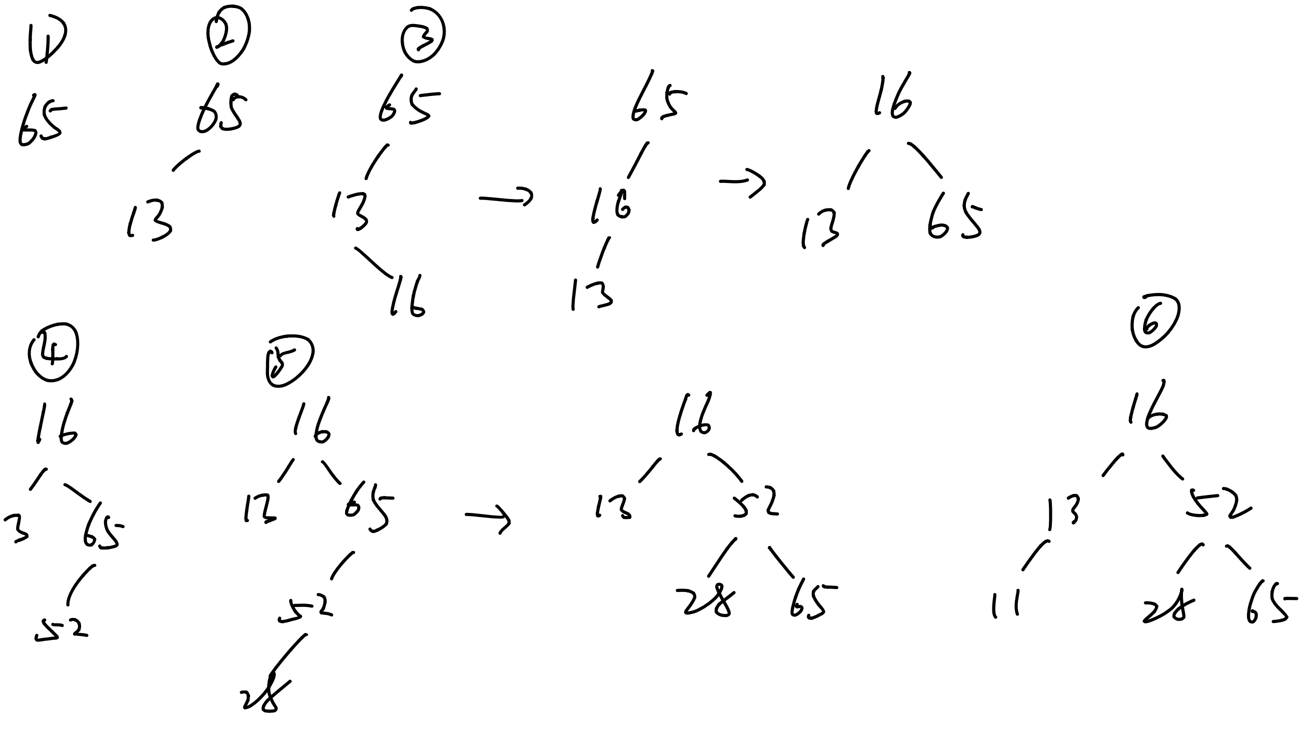
a. (2 pts) an inorder traversal
b. (2 pts) a preorder traversal
c. (2 pts) a postorder traversal

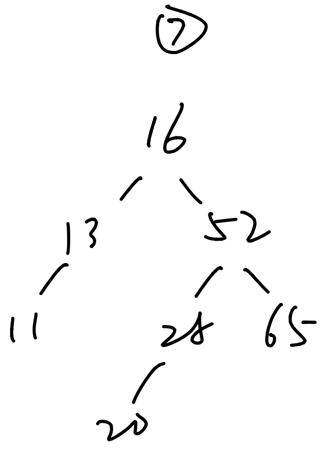
a. a – c – e – d – q – n – r – w – s
b. q – e – c – a – d – r – n – s – w
c. a – c – d – e – n – w – s – r – q

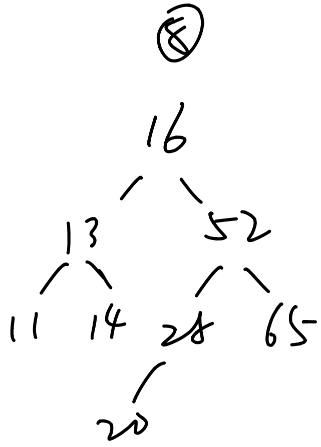
10 points

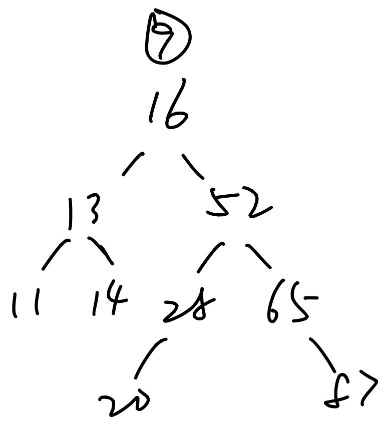
2. Draw an AVL tree for the following values inserted in this order. Illustrate each rotation that occurs:

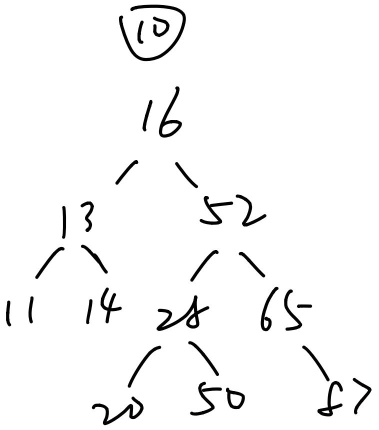
65 13 16 52 28 11 20 14 87 50 26

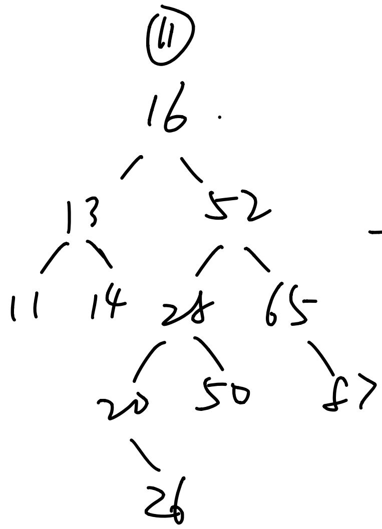


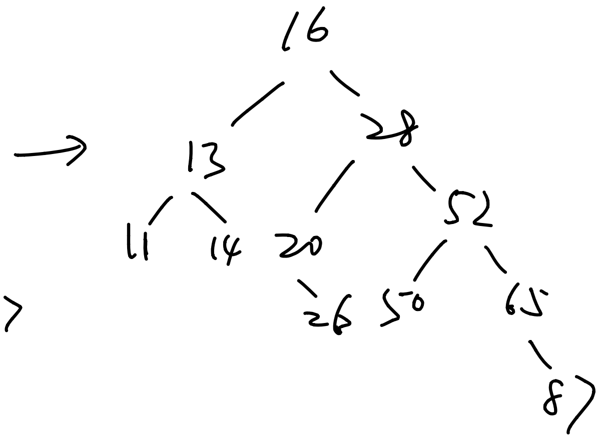


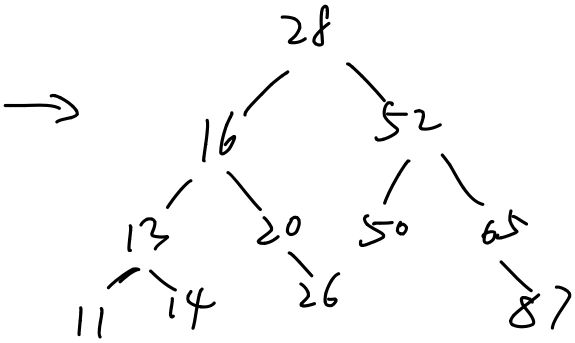








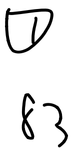


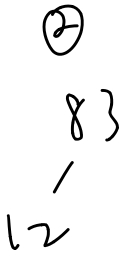


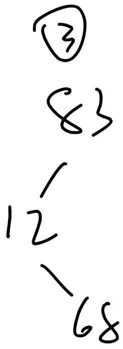
10 points

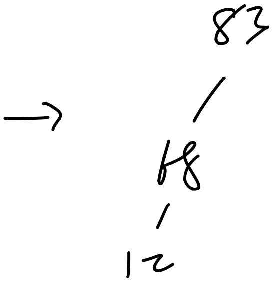
3. Draw an AVL tree for the following values inserted in this order. Illustrate each rotation that occurs:

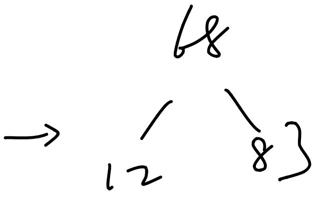
83 12 68 55 32 6 46 57 62



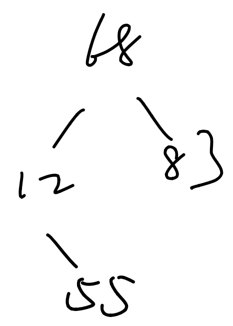




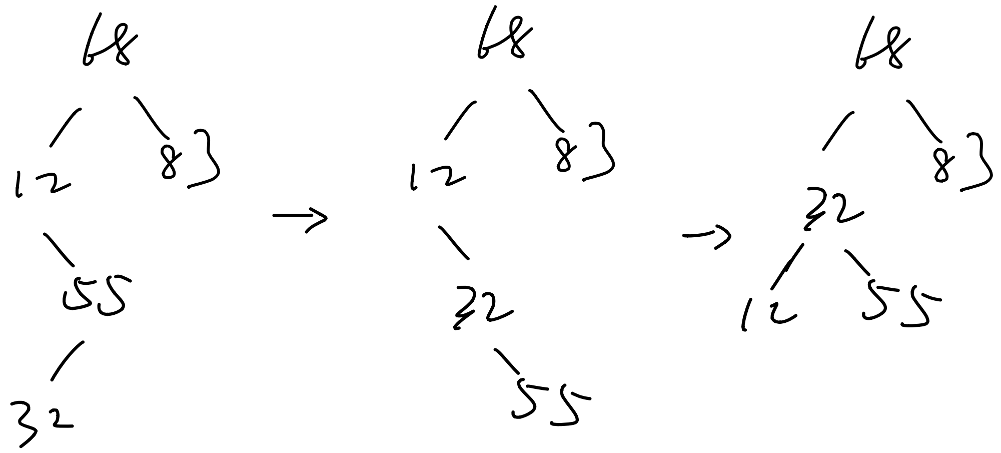




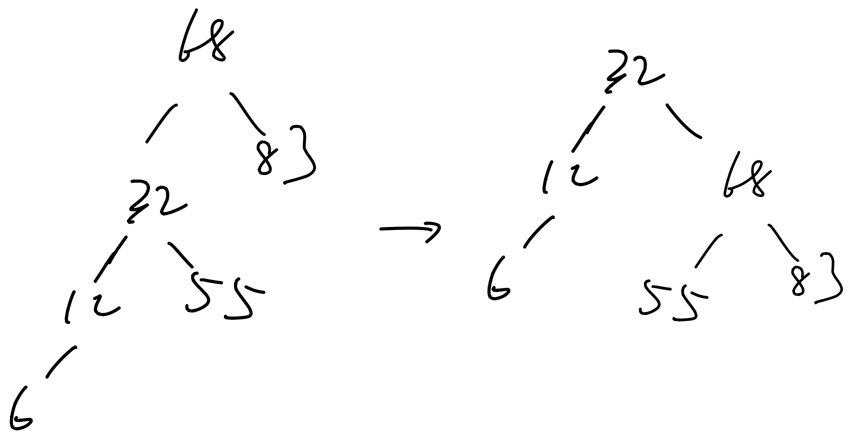
④


68
12
83
55

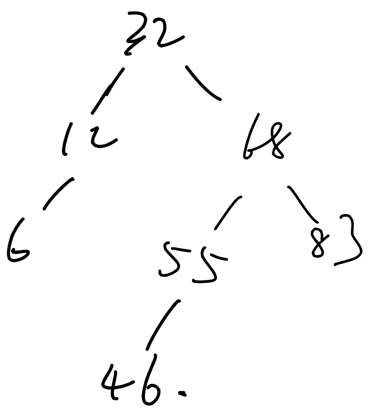
⑤


68
12
83
55
32
68
12
83
32
55
68
32
83
12
55

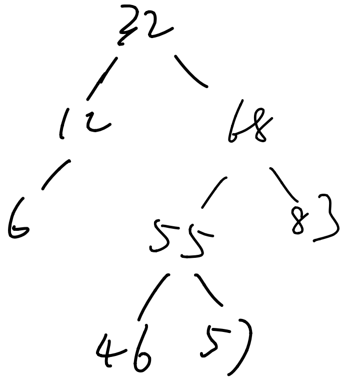
⑥


68
32
83
12
55
6
32
12
68
6
55
83

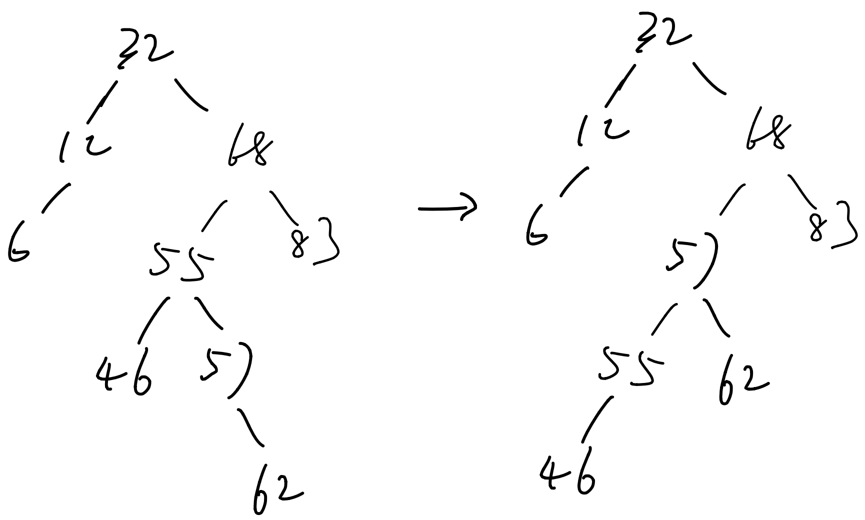
⑦

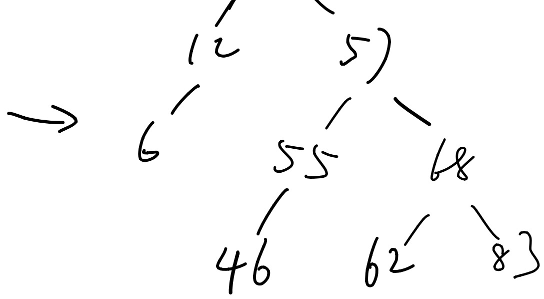

32
12
68
6
55
83
46

⑧


32
12
68
6
55
83
46
57

⑨


32
12
68
6
55
83
46
57
62
32
12
68
6
57
83
55
62
46

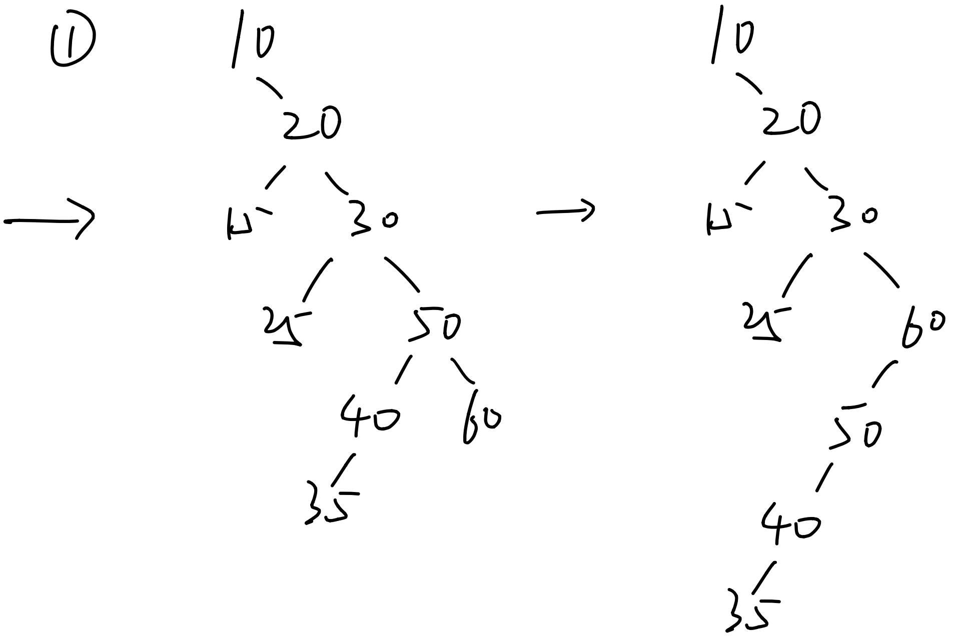

32
12
57
6
55
68
46
62
83

10 points

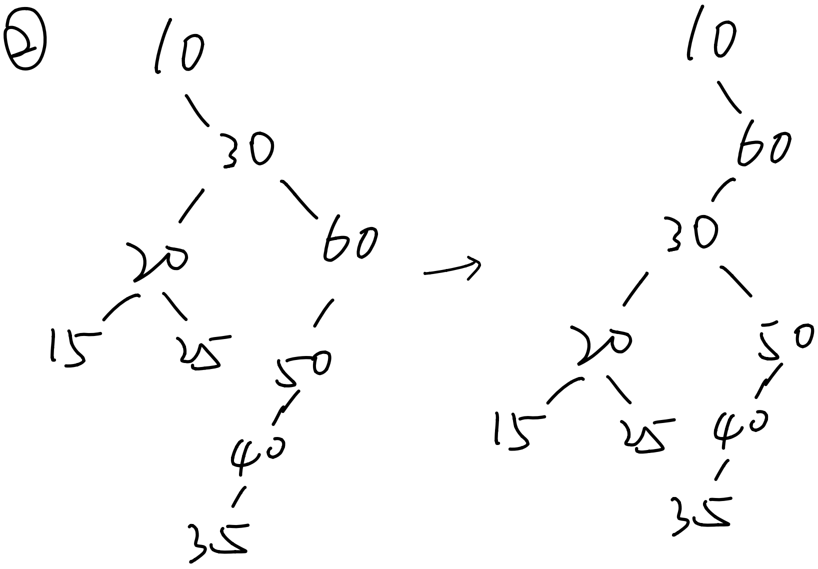
4. For the splay tree shown below, show how an access of node 60 is performed. Illustrate each operation that occurs:

```
  10
   \
    20
   /  \
  15   30
      /  \
     25   40
         /  \
        35   50
               \
                60
```

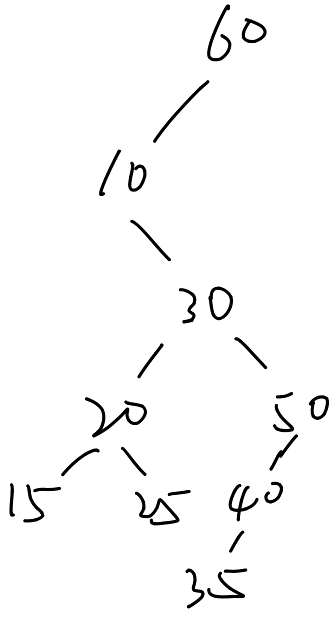
①



②



③

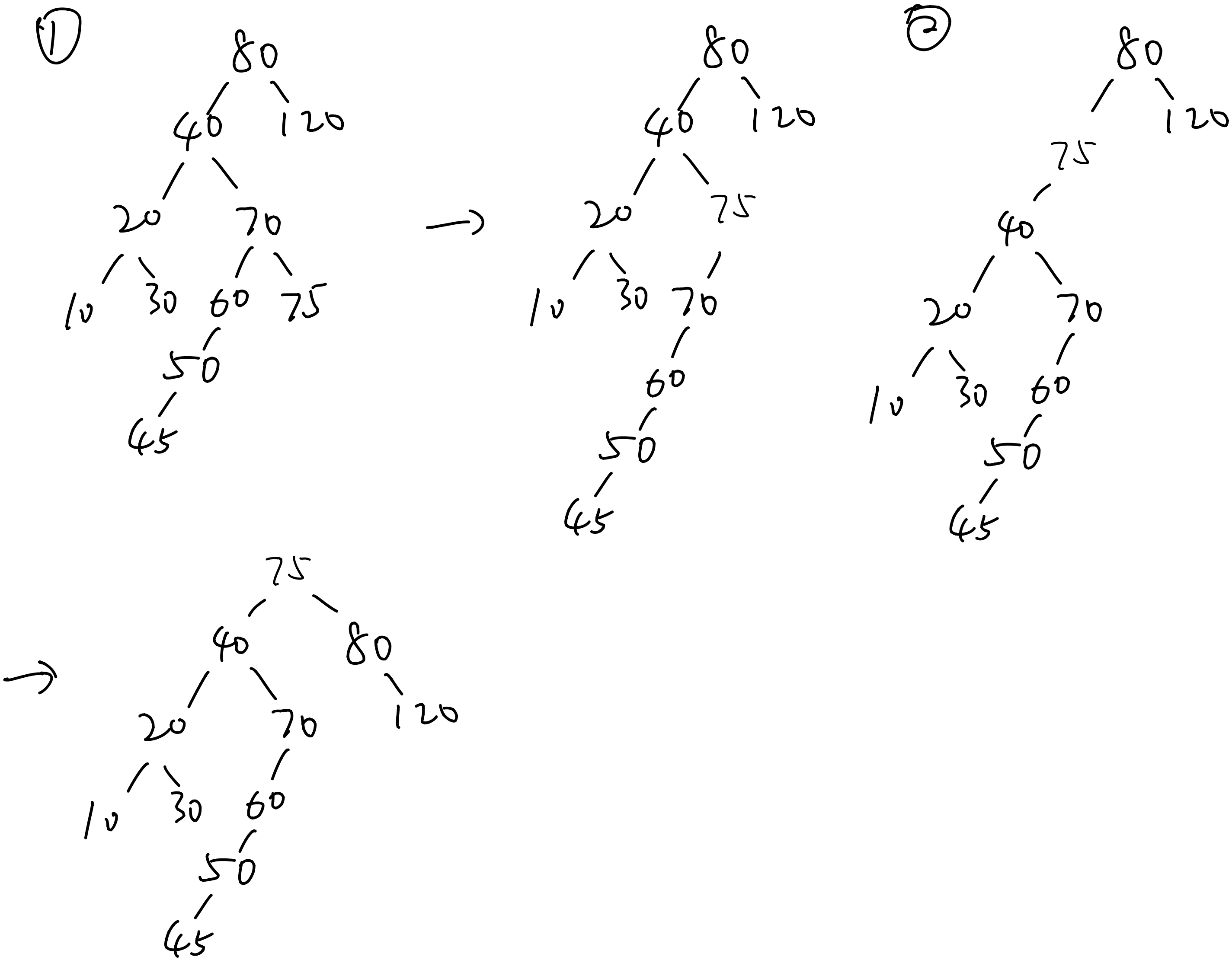


10 points

5. For the splay tree shown below, show how an access of node 75 is performed. Illustrate each operation that occurs:

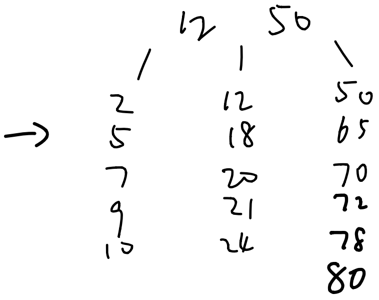
```
              80
            /     \
         40          120
        /  \
      20     60
      /\     /\
    10 30  50  70
           /     \
         45       75
```

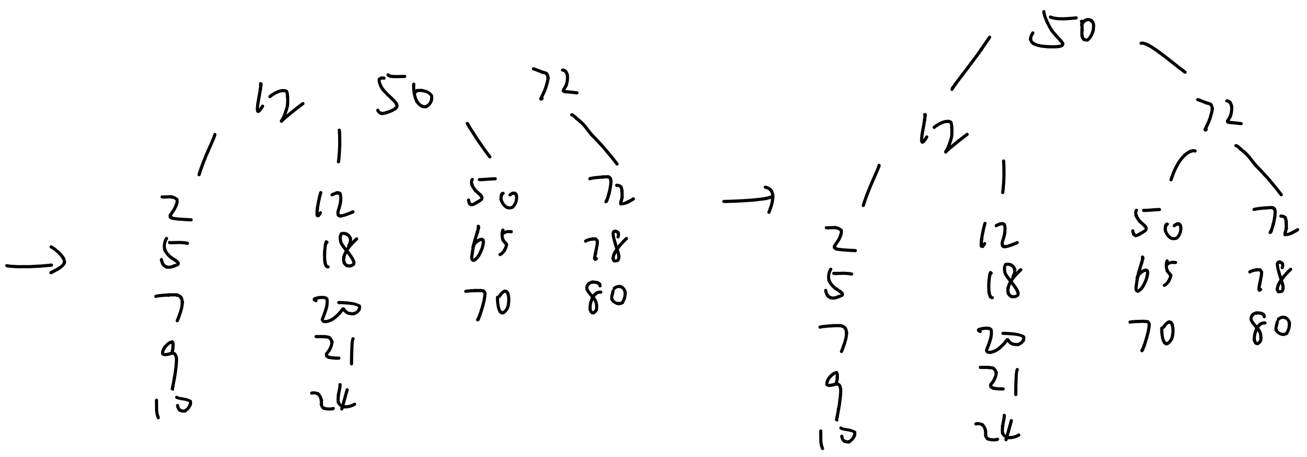
↓



10 points
6. For the B+-tree where M=3 and L=5 shown below, show how an insert of value 80 is handled.

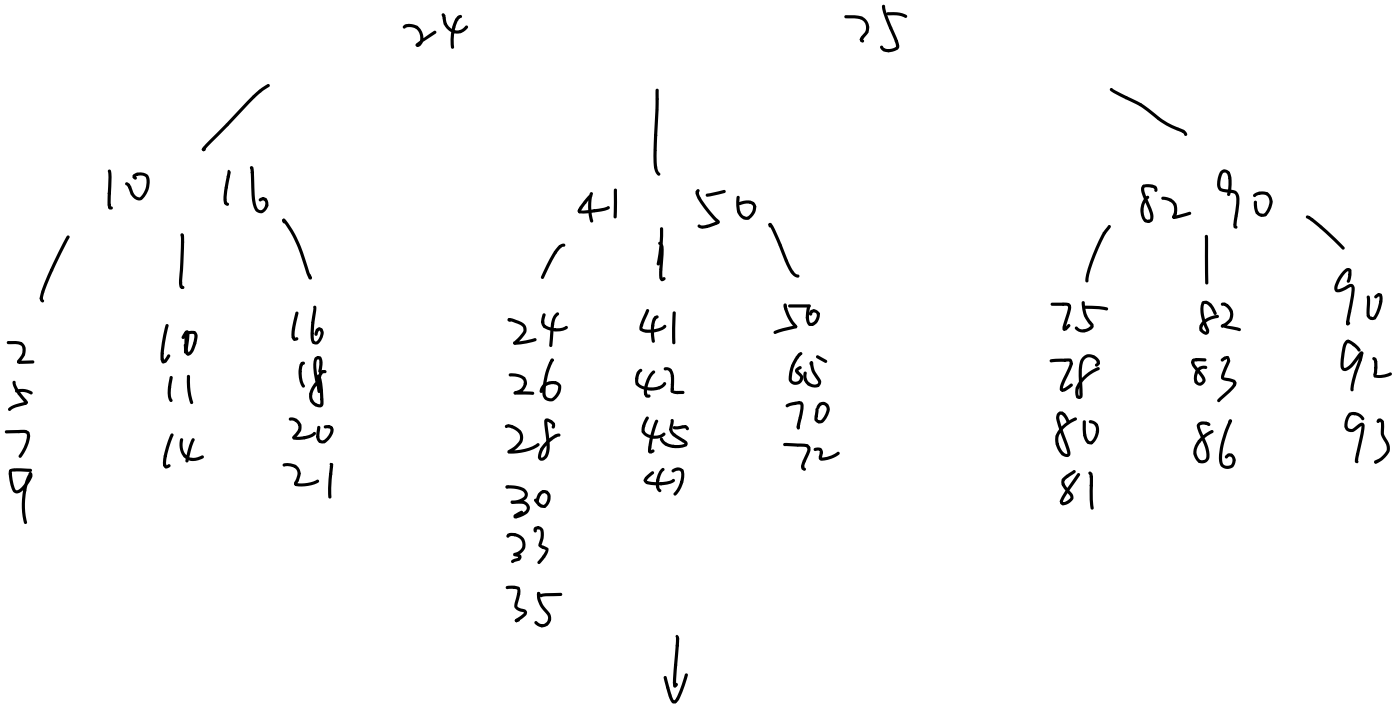
```
     || 12 || 50 ||
     /      |      \
   /        |       \
  2        12        50
  5        18        65
  7        20        70
  9        21        72
 10        24        78
```

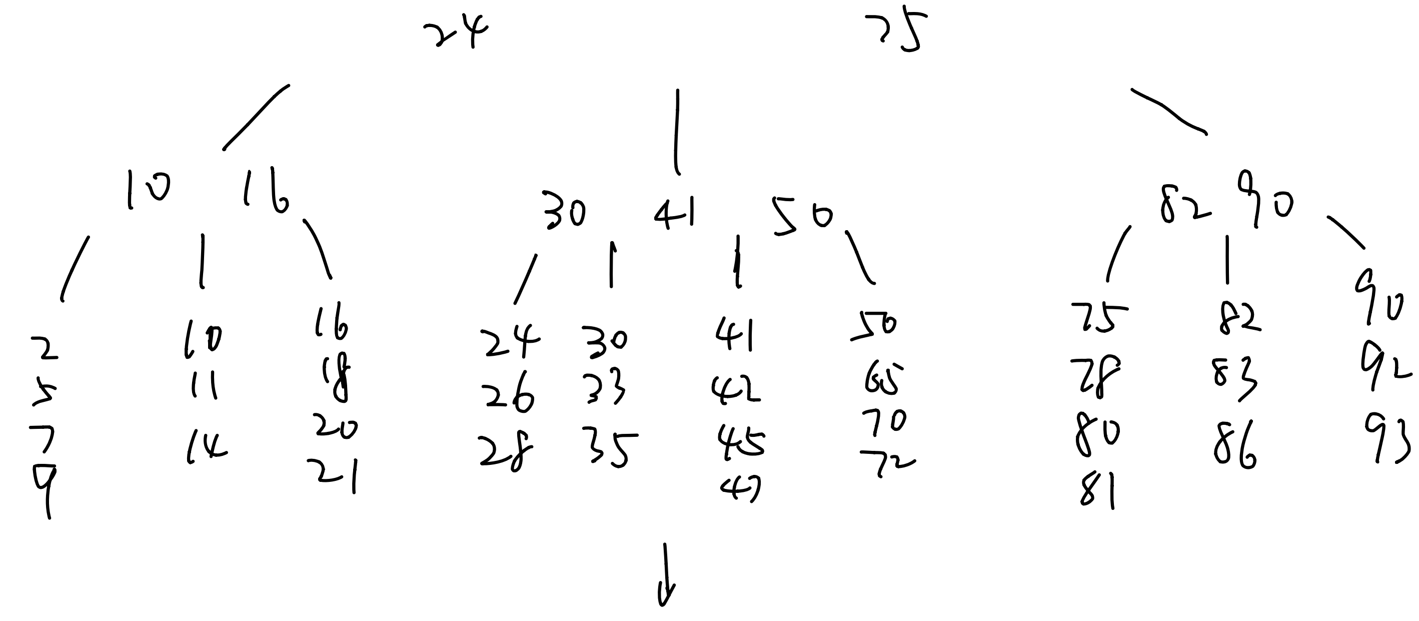


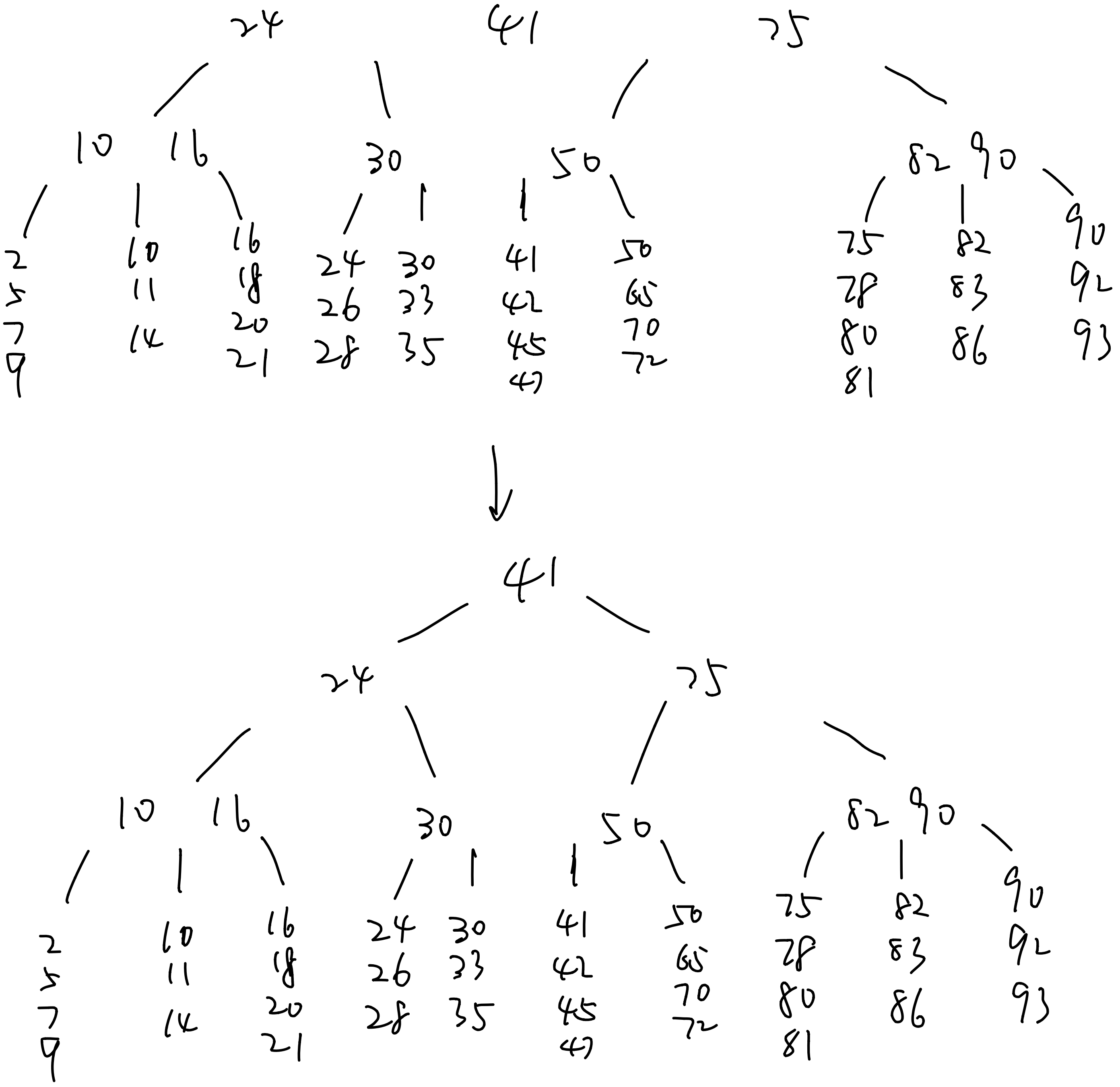


10 points
7. For the B+-tree where M=3 and L=5 shown below, show how an insert of value 28 is handled.

```
                        || 24 || 75 ||
                       /      |       \
                     /        |         \
                   /          |           \
   || 10 || 16 ||      || 41 || 50 ||       || 82 || 90 ||
    |     /      \      /     |      \       |      \      \
   /     |        |    |      |       \      |       |      |
  2      10       16   24     41       50    75      82     90
  5      11       18   26     42       65    78      83     92
  7      14       20   30     45       70    80      86     93
  9               21   33     47       72    81
                       35
```







10 points

8. A B+-tree is to be stored on disk whose block size is 3096 bytes. The data records to be stored are 36 bytes, and their key is 4 bytes. Determine the values for M and L for the B+-tree. Assume pointers are 4 bytes each.

$(M-1)\times 4 + M\times 4 = 3096 \Rightarrow M = 387.5 \Rightarrow M = 387$

$L = 3096 / 36 = 86$

8 points

9. For the problem above, how many levels are needed to store 8,600,000 records?

In the worst case we can have

$8600000 / (86/2) = 200000$ leaves.

$\lceil 387/2 \rceil = \lceil 193.5 \rceil = 194$ branches.

The level $= \lceil \log_{194} 200000 + 1 \rceil = \lceil 3.3 \rceil = 4$ levels

8 points

10. If a binary tree has N nodes, how many null child pointers will it have? Explain your reasoning.

It will have N+1 null pointers.

Adding a Node takes one position and create two position.

Initially we have nodeCount = 1, nullPointerCount = 2.

each time we insert a node, nodeCount = nodeCount + 1

Also, nullPointerCount = nullPointerCount − 1 + 2
= nullPointerCount + 1

Thus at any time, the difference between nullpointerCount and nodeCount is 1. thus we will have N+1 null pointers in a binary tree.

8 points
11. In a perfect binary tree (one filled at every level), what does adding another level do to the number of nodes in the tree?

Suppose we have a perfect binary tree of Node N and level of N_level.

Thus: $N = 2^{N\_level} - 1$

Then adding another level will create 2^{N_level} number of nodes to the tree. Thus we will have $N = N + 2^{N\_level} = N + N + 1 = 2N+1$ Nodes.