**Advanced Algorithms**

**Exercise for Lecture 6**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Name** |  | **Student ID** |  |
| **Problem 1** |  | | |
| **Problem 2** |  | | |
| **Problem 3** |  | | |
| **Total Score** |  | | |
| **Notes** | Deadline: **2023-10-03 24:00**  Submission Format: ‘**Lecture6\_Name\_Student ID.docx**’, and please send to: **[chenlq1997@126.com](mailto:algorithms_23fall@163.com)**.  This assignment is meant to be an evaluation of your **individual** understanding coming into the course and should be completed **without collaboration** or outside help. | | |

**Problem 1.[20 points]**

If we use the RB-INSERT to insert a key k into a red black tree, and then use RB-DELETE to delete the key k from the red black tree. Is the latter red black tree sure to be the same with the former red black tree? Please briefly explain the reason and give an example.

**Solution:**

They can be different.

Because when we insert the key, the tree is likely to change the structure. So when we delete the key, if the structure doesn’t change, the latter red black tree will be different from the former red black tree.

The example is shown below.

2

2

1

**Insert 3 Delete 3**

1

1

3

2

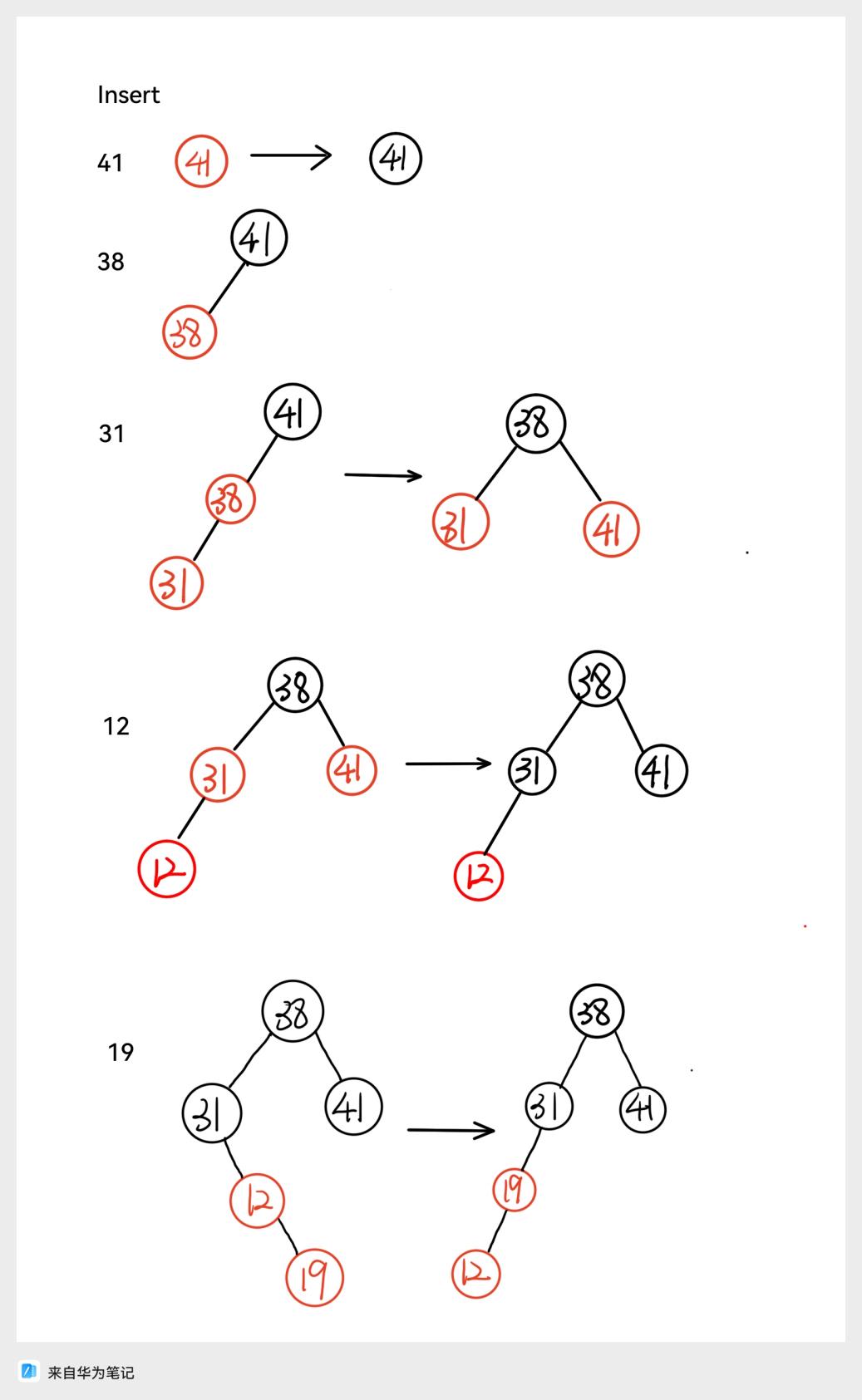
**Problem 2.[40 points]**

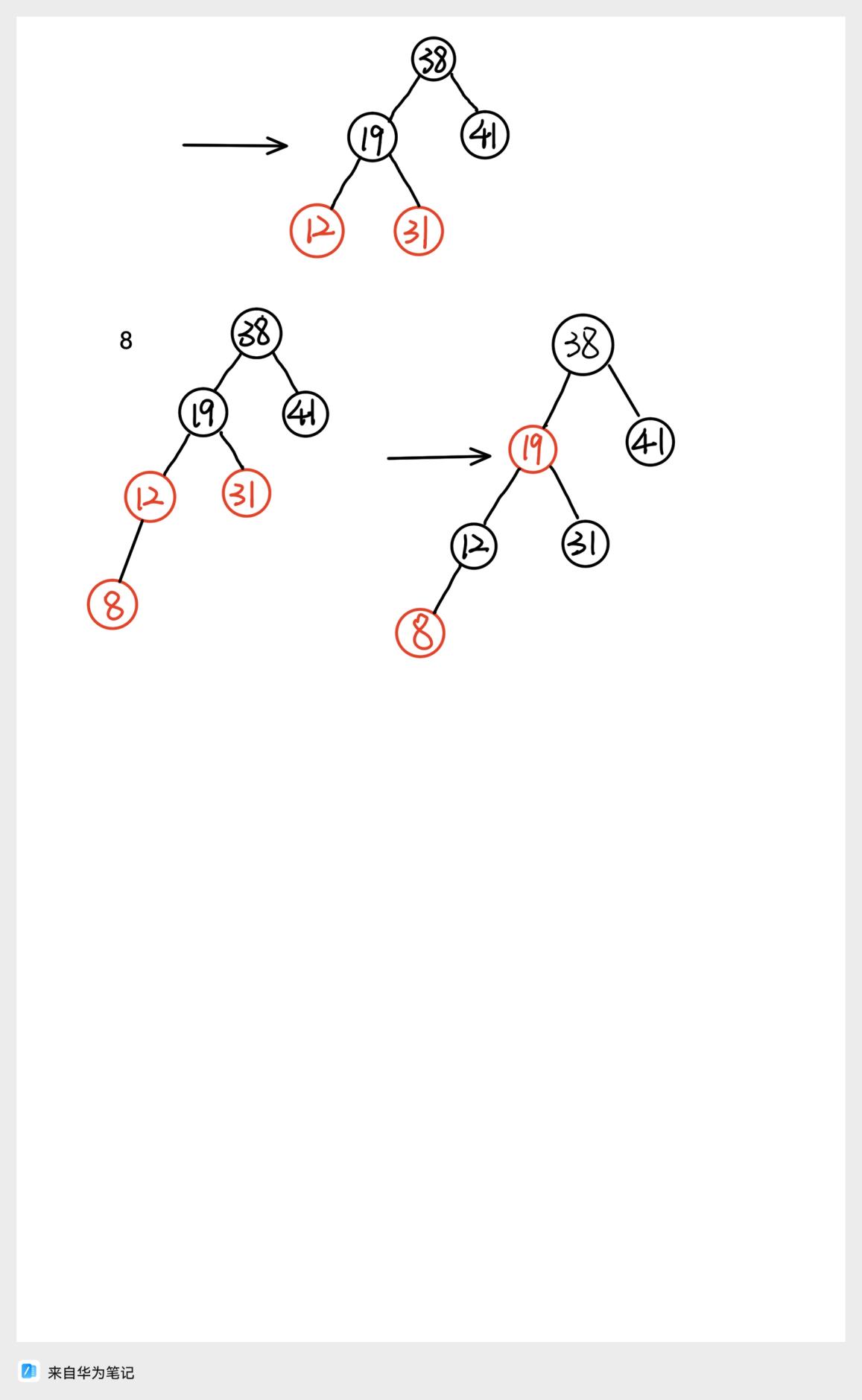
2.1 Please insert key value 41, 38, 31, 12, 19, 8 into a red black tree one by one. Please show each intermediate state of the red black tree.

2.2 Please delete key value 8, 12, 19, 31, 38, 41 from the red black tree one by one which you get in problem 2.1. Please show each intermediate state of the red black tree.

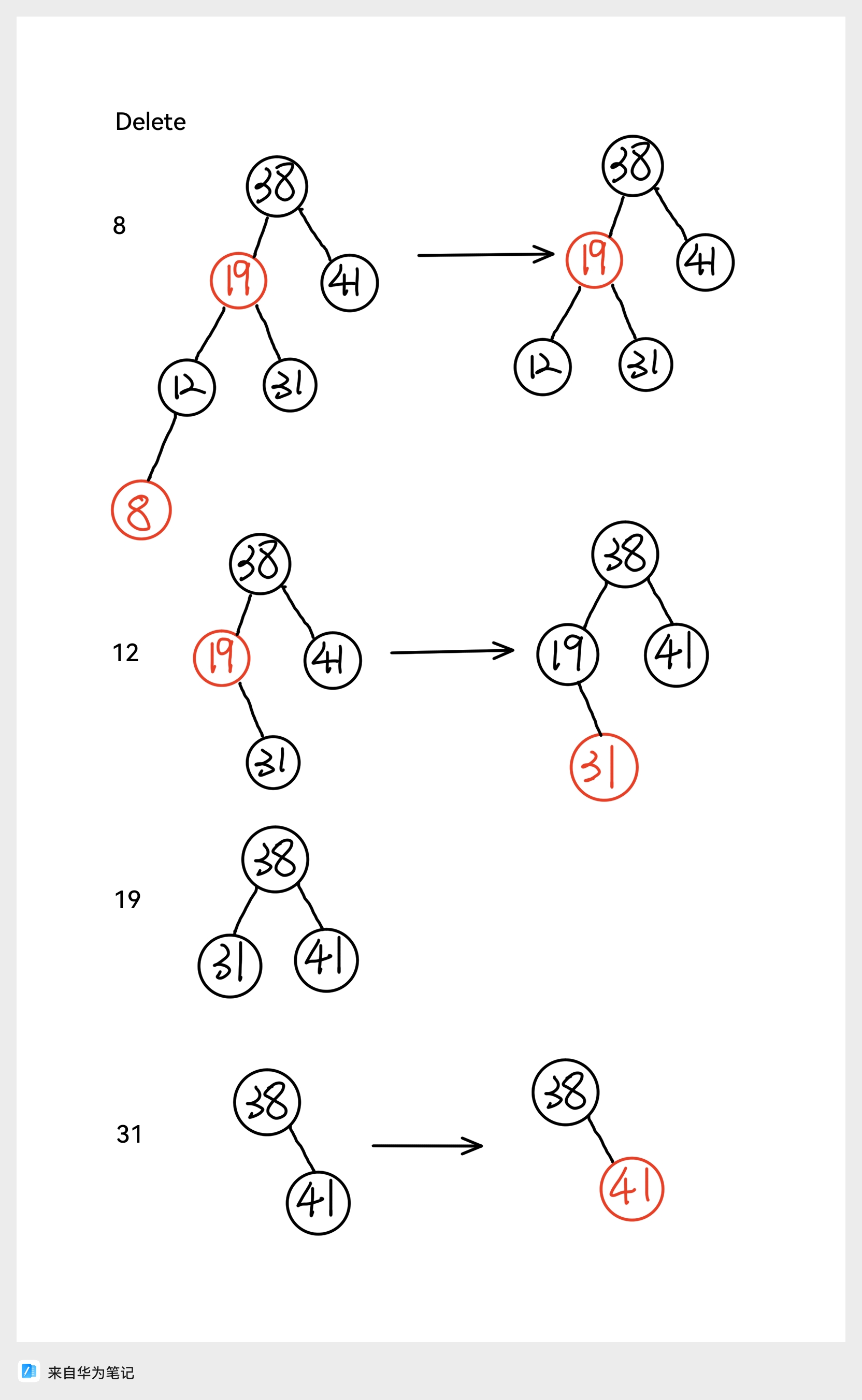
**Solution:**

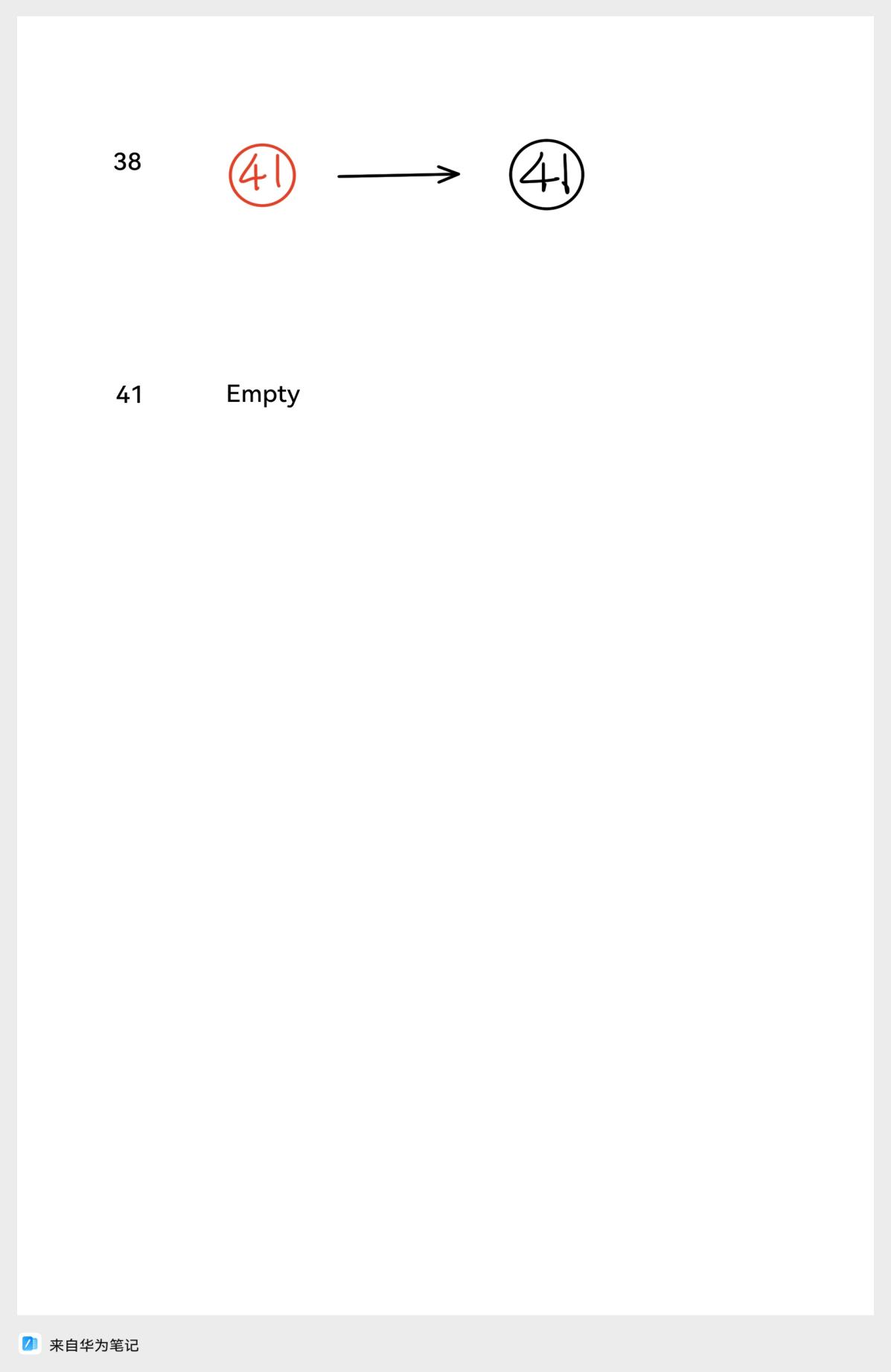
1.1





1.2





**Problem 3.[40 points]**

As is shown in picture one, it is the code of RB-DELETE-FIXUP(T, z), which is used to balance the black height of the red black tree after the operation of deletion. Please prove that the root of red black tree is sure to be black after the implementation of RB-DELETE-FIXUP(T, z). You should give the reason for each case in the code.

RB-DELETE-FIXUP(T,x)

1 while x != T.root and x.color == BLACK

2 if x == x.p.left

3 w = x.p.right

4 if w.color == RED

5 w.color = BLACK //case 1

6 x.p.color = RED //case 1

7 LEFT-ROTATE(T, x, p) //case 1

8 w = x.p.right //case 1

9 if w.left.color == BLACK andw.right.color == BLACK

10 w.color = RED //case 2

11 x = x.p //case 2

12 else if w.right.color == BLACK

13 w.left.color = BLACK //case 3

14 w.color = RED //case 3

15 RIGHT-ROTATE(T, w) //case 3

16 w = x.p.right //case 3

17 w.color = x.p.color //case 4

18 x.p.color = BLACK //case 4

19 w.right.color = BLACK //case 4

20 LEFT-ROTATE(T, x, p) //case 4

21 x = T.root //case 4

22 else(same as then clause with “right”and “left” exchanged)

23 x.color = BLACK

Picture 1. The code of R-DELETE-FIXUP(T, z)

**Solution:**

case 1：In the code we can find that the code won’t exit the while loop from case 1. The code is sure to enter case 2, 3 or 4. So we just need to discuss the case 2, 3 and 4.

case 2：If the code exit the loop from case 2, the x = T.root. It means that the code in line 23, x.color = BLACK, will make T.root.color equal BLACK.

case 3：It is sure to enter case 4. So it is not important.

case 4：In line 21, x=T.root, the code will exit the loop. And in line 23, x.color = BLACK, it will make the T.root.color = BLACK.

Therefore, The root of red black tree will be black.