

二叉树.

1. 176

2. 分支节点有: 1, 2, 3, 4, 5, 7

叶子节点有: 6, 8, 9.

注: - 后面对应该结点的层次

1-1, 2-2, 3-2, 4-3, 5-3, 6-3, 7-4, 8-4, 9-5

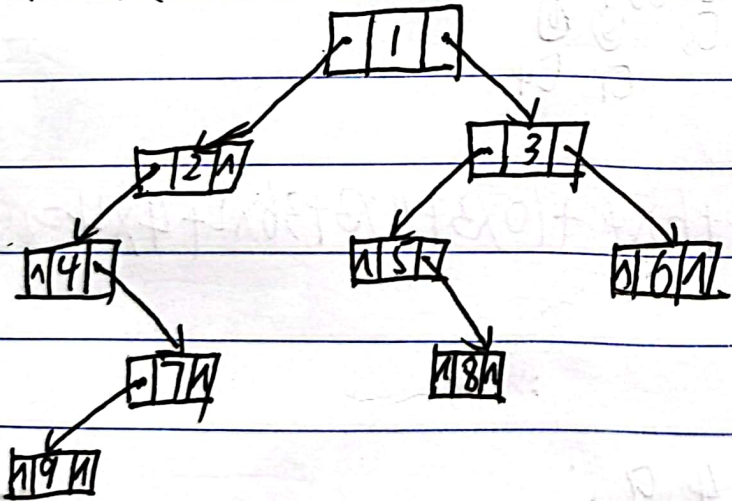
3.

||| 顺序表示

1 2 3 4 \emptyset 5 6 \emptyset 7 \emptyset \emptyset \emptyset 8 \emptyset \emptyset \emptyset 9.

数组 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17.

二叉链表表示



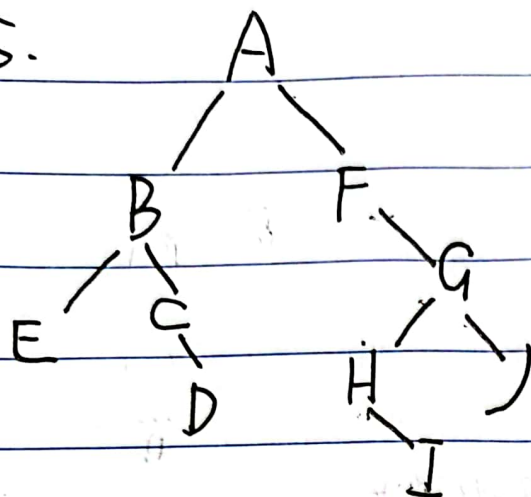
4. 先序遍历: 124793586

中序遍历: 497215836

后序遍历: 974285631



5.



6. $C_1: 0110$

$C_2: 10$

$C_3: 0000$

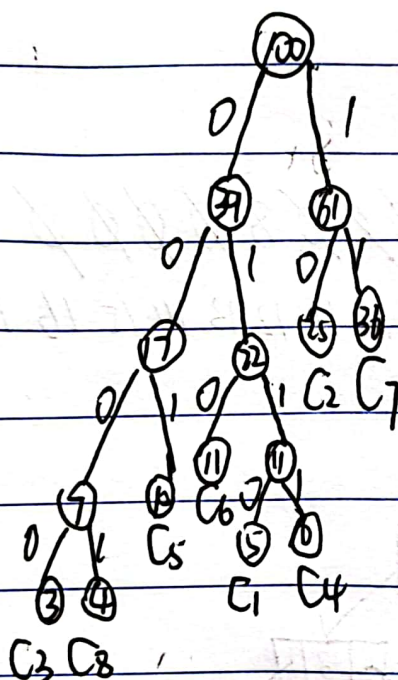
$C_4: 0111$

$C_5: 001$

$C_6: 010$

$C_7: 11$

$C_8: 0001$



$$WPL = 5 \times 4 + 25 \times 2 + 3 \times 4 + 6 \times 4 + 10 \times 3 + 11 \times 3 + 36 \times 2 + 4 \times 4 = 257$$

总码数为257.

7. 不是, 调整后为:

64 45 24 11 7 18 4 9.

第一次输出后状态:

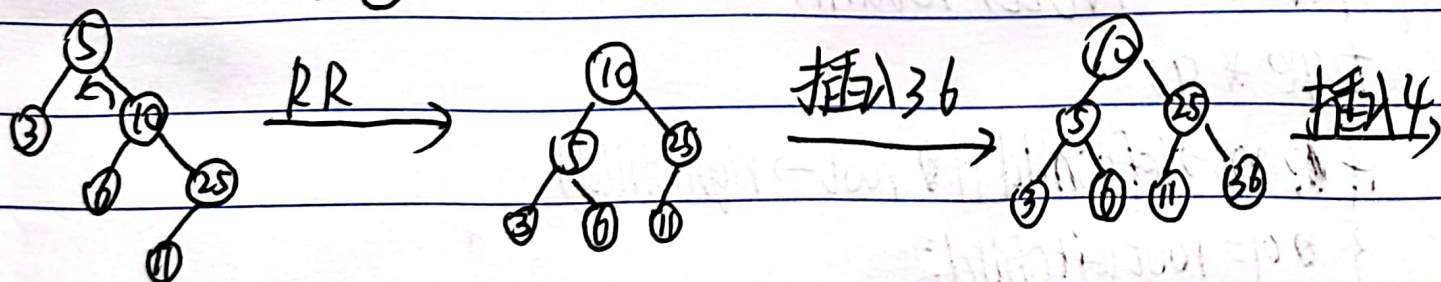
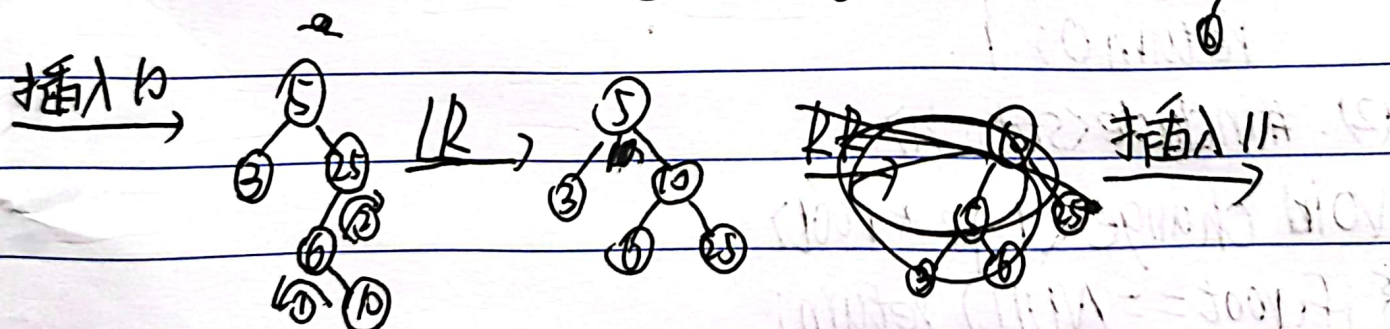
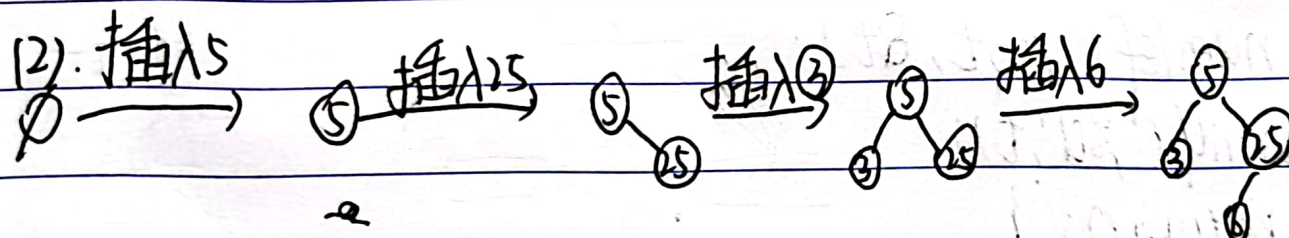
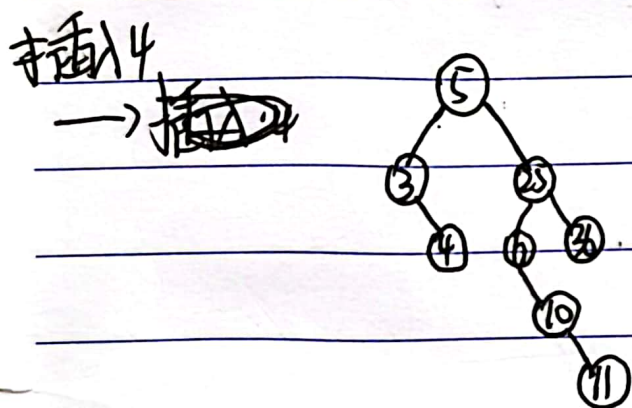
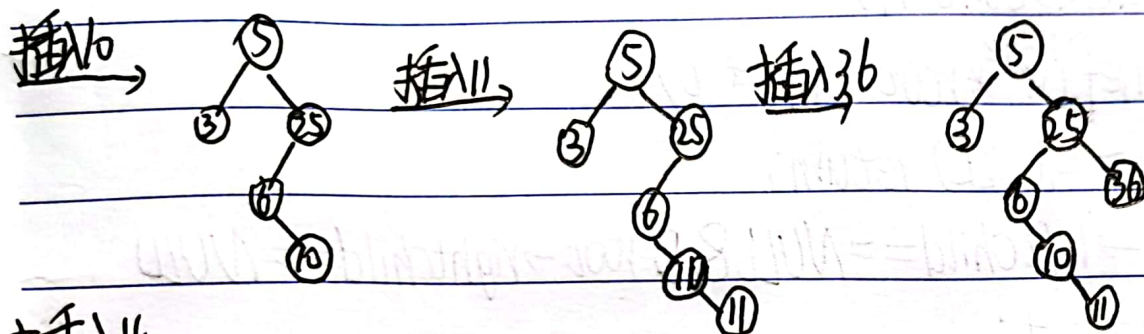
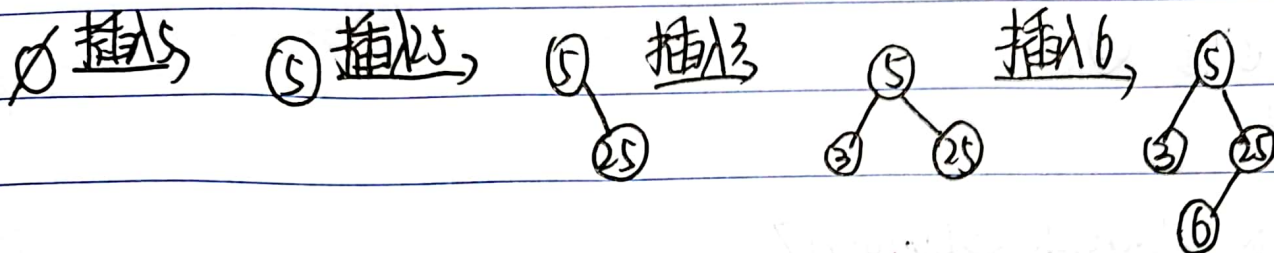
45 11 24 9 7 18 4

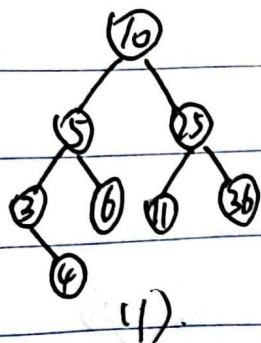
第二次输出后状态:

24 11 18 9 7 4



11). ~~插入5~~





9. ~~*~~ #include <stdio.h>

```
void numlef(type *root, int *t).
```

```
{ if (root == NULL) return;
```

```
  if (root->leftchild == NULL && root->rightchild == NULL)
    (*t)++;
```

```
  numlef(root->leftchild, t);
```

```
  numlef(root->rightchild, t); }
```

```
int main()
```

```
{ int t,
```

```
  numlef(root, &t);
```

```
  printf("%d", t);
```

```
  return 0; }
```

(2). #include <stdio.h>

```
void change <type *root>
```

```
{ if (root == NULL) return;
```

```
  type *q;
```

```
  if (root->leftchild || root->rightchild)
```

```
  { *q = root->leftchild;
```




```
root->leftchild = root->rightchild;
```

```
root->right = q; }
```

```
change(root->leftchild);
```

```
change(root->rightchild);
```

```
int main( )
```

```
{ change(root);
```

```
  return 0 ; }
```

