




# buuctf LeftOrRight

原创

是真的白  于 2022-04-16 22:13:45 发布  267  收藏

文章标签: [密码学](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: [https://blog.csdn.net/m0\\_62506844/article/details/123648874](https://blog.csdn.net/m0_62506844/article/details/123648874)

版权

```
ugh2.jpg x
00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
00 30 78 36 36 0d 0a 30 78 33 30 0d 0a 30 78 33 39 0x 6 6 . . 0 x 3 0 . . 0 x 3 9
0d 0a 30 78 36 35 0d 0a 30 78 33 35 0d 0a 30 78 . . 0 x 6 5 . . 0 x 3 5 . . 0 x
33 34 0d 0a 30 78 36 33 0d 0a 30 78 33 31 0d 0a 34 . . 0 x 6 3 . . 0 x 3 1 . .
30 78 36 32 0d 0a 30 78 36 31 0d 0a 30 78 36 34 0 x 6 2 . . 0 x 6 1 . . 0 x 6 4
0d 0a 30 78 33 32 0d 0a 30 78 37 38 0d 0a 30 78 . . 0 x 3 2 . . 0 x 7 8 . . 0 x
33 33 0d 0a 30 78 33 38 0d 0a 30 78 36 64 0d 0a 33 3 . . 0 x 3 8 . . 0 x 6 d . .
30 78 37 36 0d 0a 30 78 37 39 0d 0a 30 78 36 37 0 x 7 6 . . 0 x 7 9 . . 0 x 6 7
0d 0a 30 78 33 37 0d 0a 30 78 37 37 0d 0a 30 78 . . 0 x 3 7 . . 0 x 7 7 . . 0 x
37 61 0d 0a 30 78 36 63 0d 0a 30 78 37 33 0d 0a 7 a . . 0 x 6 c . . 0 x 7 3 . .
30 78 37 35 0d 0a 30 78 36 38 0d 0a 30 78 36 62 0 x 7 5 . . 0 x 6 8 . . 0 x 6 b
0d 0a 30 78 36 39 0d 0a 30 78 36 61 0d 0a 30 78 . . 0 x 6 9 . . 0 x 6 a . . 0 x
36 65 0d 0a 30 78 36 66 0d 0a 30 78 37 30 ff d8 6 e . . 0 x 6 f . . 0 x 7 0 ?
ff e0 00 10 4a 46 49 46 00 01 01 01 00 60 00 60 ? . . J F I E
00 00 ff db 00 43 00 02 01 01 02 01 01 02 02 02 . . ? . C . . . . . . . .
```

```
00a7e0 40 05 14 51 40 05 14 51 40 05 14 51 40 05 14 51 e . . Q @ . . Q @ . . Q @ . . Q
00a7f0 40 05 14 51 40 05 14 51 40 05 14 51 40 05 14 51 @ . . Q @ . . Q @ . . Q @ . . Q
00a800 40 05 14 51 40 05 14 51 40 05 14 51 40 05 14 51 @ . . Q @ . . Q @ . . Q @ . . Q
00a810 40 05 14 51 40 05 14 51 40 05 14 51 40 1f ff d9 @ . . Q @ . . Q @ . . Q @ . ?
00a820 30 78 33 39 0d 0a 30 78 33 30 0d 0a 30 78 33 35 x 3 9 . . 0 x 3 0 . . 0 x 3 5
00a830 0d 0a 30 78 36 35 0d 0a 30 78 33 34 0d 0a 30 78 . . 0 x 6 5 . . 0 x 3 4 . . 0 x
00a840 36 33 0d 0a 30 78 33 31 0d 0a 30 78 36 36 0d 0a 6 3 . . 0 x 3 1 . . 0 x 6 6 . .
00a850 30 78 36 31 0d 0a 30 78 37 38 0d 0a 30 78 33 33 0 x 6 1 . . 0 x 7 8 . . 0 x 3 3
00a860 0d 0a 30 78 33 32 0d 0a 30 78 33 38 0d 0a 30 78 . . 0 x 3 2 . . 0 x 3 8 . . 0 x
00a870 36 64 0d 0a 30 78 36 34 0d 0a 30 78 37 39 0d 0a 6 d . . 0 x 6 4 . . 0 x 7 9 . .
00a880 30 78 37 36 0d 0a 30 78 36 37 0d 0a 30 78 33 37 0 x 7 6 . . 0 x 6 7 . . 0 x 3 7
00a890 0d 0a 30 78 37 37 0d 0a 30 78 36 32 0d 0a 30 78 . . 0 x 7 7 . . 0 x 6 2 . . 0 x
00a8a0 37 33 0d 0a 30 78 37 35 0d 0a 30 78 36 38 0d 0a 7 3 . . 0 x 7 5 . . 0 x 6 8 . .
00a8b0 30 78 36 62 0d 0a 30 78 36 63 0d 0a 30 78 36 39 0 x 6 b . . 0 x 6 c . . 0 x 6 9
00a8c0 0d 0a 30 78 36 61 0d 0a 30 78 37 61 0d 0a 30 78 . . 0 x 6 a . . 0 x 7 a . . 0 x
00a8d0 36 65 0d 0a 30 78 36 66 0d 0a 30 78 37 30 6 e . . 0 x 6 f . . 0 x 7 0 . .
00a8e0 . . . . . . . . . . . . . . . . . . . . . . . . . .
00a8f0 . . . . . . . . . . . . . . . . . . . . . . . . . .
00a900 . . . . . . . . . . . . . . . . . . . . . . . . . .
00a910 . . . . . . . . . . . . . . . . . . . . . . . . . .
00a920 . . . . . . . . . . . . . . . . . . . . . . . . . .
```

发现两段16进制数据

转换为ascii为f09e54c1bad2x38mvyg7wzlsuhkijnop 和905e4c1fax328mdyvg7wsuhklijznop

提示为查找右边的, 应该就是二叉树算法中的后序

百度搜一下二叉树算法好啦

```

# f09e54c1bad2x38mvyg7wzlsuhkijnop
# 905e4c1fax328mdyvg7wbsuhklijznop

def fromFMtoL( mid ):
    global las #全局后序遍历
    global fir #先序遍历
    root = fir[0] #取出当前树根
    fir = fir[1:] #取出树根后 先序遍历把根拿出来 下面一个元素做树根
    root_po = mid.find( root ) #在中序遍历当中树根的位置
    left = mid[0:root_po] #左子树
    right = mid[root_po+1:len(mid)] #右子树
    ...

    后序遍历: 左 右 根
    先左子树 再右子树 最后跟
    ...

    #有左子树的时候
    if len(left) > 0:
        fromFMtoL( left )
    #有右子树的时候
    if len(right) > 0:
        fromFMtoL( right )
    #树根写进结果
    las += root

if __name__ == "__main__" :
    # fir = input("请输入先序遍历: ") #前序遍历的结果
    # mid = input("请输入中序遍历: ") #中序遍历的结果
    fir = "f09e54c1bad2x38mvyg7wzlsuhkijnop"
    mid = "905e4c1fax328mdyvg7wbsuhklijznop"
    # fir = "ABC"
    # mid = "BAC"
    las = ""
    fromFMtoL( mid )
    print(las)

```