

C++方向编程题答案

第二周

day11

题目ID: 24979-最近公共祖先

链接: <https://www.nowcoder.com/practice/70e00e490b454006976c1fdf47f155d9?tpId=8& tqId=11017&rp=1&ru=/activity/oj&qr=/ta/cracking-the-coding-interview/question-ranking>

【题目解析】:

最近公共祖先表示距离两个节点最近的公共父节点, 这道题考察二叉树。

【解题思路】:

题目所描述的满二叉树如下: 1 /\ 2 3 /\ /\ 4 5 6 7 上述树中子节点与父节点之间的关系为 $root = child / 2$ 所以如果 $a \neq b$, 就让其中的较大数除以2, 如此循环直到 $a == b$ 即是原来两个数的最近公共祖先 比如: 2 和7的最近公共祖先: $7/2 = 3 \rightarrow 3/2 = 1, 2/2 = 1$, 得到1为它们的公共祖先

```
class LCA {
public:
    int getLCA(int a, int b) {
        while (a != b)
        {
            if (a > b)
                a /= 2;
            else
                b /= 2;
        }
        return a;
    }
};
```

题目ID: 36910-求最大连续bit数

链接: <https://www.nowcoder.com/practice/4b1658fd8ffb4217bc3b7e85a38cfaf2?tpId=37& tqId=21309&rp=1&ru=/activity/oj&qr=/ta/huawei/question-ranking>

【题目解析】:

这道题考察位运算

【解题思路】:

根据位运算, 获取每一位的二进制值。获取第i位的值: $(n >> i) \& 1$ 。如果1连续, 则计数累加, 如果不连续, 则从0开始计数。

```
#include<iostream>
#include<string>
#include<algorithm>
```

```
using namespace std;
int main()
{
    int n;
    while (cin >> n)
    {
        int count = 0, maxCount = 0;
        // n右移32次就变为0
        while (n)
        {
            //获取当前位的二进制值
            if (n & 1)
            {
                //如果1的值连续，计数累加，并且跟新最大计数
                ++count;
                maxCount = max(count, maxCount);
            }
            else
            {
                //如果1的值不连续，重新累加
                count = 0;
            }
            //右移一次，为获取下一位二进制值做准备
            n = n >> 1;
        }
        cout << maxCount << endl;
    }
    return 0;
}
```