

编写一个算法来判断一个数是不是“快乐数”。

一个“快乐数”定义为：对于一个正整数，每一次将该数替换为它每个位置上的数字的平方和，然后重复这个过程直到这个数变为 1，也可能是无限循环但始终变不到 1。如果可以变为 1，那么这个数就是快乐数。

**示例：**

**输入：** 19

**输出：** true

**解释：**

$1^2 + 9^2 = 82$

$8^2 + 2^2 = 68$

$6^2 + 8^2 = 100$

$1^2 + 0^2 + 0^2 = 1$

思路，按照数来计算，把出现过的数用map统计起来，若是重复出现，那么就进入循环，不是快乐数，若是跳出循环，那么是快乐数。

```
class Solution {
public:
    bool isHappy(int n) {
        map<int,int> bj;
        while(n>1){
            int i=n;
            int max=0;
            while(i>0){
                max=max+(i%10)*(i%10);
                i=i/10;
            }
            n=max;
            if(bj[max]==1){
                return false;
            }
            bj[max]=1;
        }
        return true;
    }
};
```