

丑数

我们把只包含质因子 2、3 和 5 的数称作丑数（Ugly Number）。求按从小到大的顺序的第 n 个丑数。

示例：

输入： $n = 10$

输出：12

解释：1, 2, 3, 4, 5, 6, 8, 9, 10, 12 是前 10 个丑数。

方法一：最小堆

要得到从小到大的第 n 个丑数，可以使用**最小堆**实现。

初始时堆为空。首先将最小的丑数 1 加入堆。

每次取出堆顶元素 x ，则 x 是堆中最小的丑数，由于 $2x, 3x, 5x$ 也是丑数，因此将 $2x, 3x, 5x$ 加入堆。

上述做法会导致堆中出现重复元素的情况。为了避免重复元素，可以使用哈希集合去重，避免相同元素多次加入堆。

在排除重复元素的情况下，第 n 次从最小堆中取出的元素即为第 n 个丑数。

```
class Solution {
public:
    int nthUglyNumber(int n) {
        vector<int> bases={2,3,5};
        priority_queue<long,vector<long>,greater<long>> heap; //从小到大
        unordered_set<long> s;
        int target=0;
        heap.push(1);
        s.insert(1);
        for(int i=0;i<n;i++)
        {
            long cur=heap.top();
            heap.pop();
            target=(int)cur;
            for(auto& b:bases)
            {
                long next=cur*b;
                if(!s.count(next))//查看是否有重复元素，无重复则添加
                {
                    s.insert(next);
                }
            }
        }
    }
};
```

```
        heap.push(next);
    }
}
return target;
}
```

```
};
```