剑指49 丑数

方法一

最小堆原来不用实现，直接用c++的优先级队列！！

```cpp
class Solution {
public:
    int nthUglyNumber(int n) {
        vector<int> factors = {2, 3, 5};
        unordered_set<long> seen;
        priority_queue<long, vector<long>, greater<long>> heap;
        seen.insert(1L);
        heap.push(1L);
        int ugly = 0;
        for (int i = 0; i < n; i++) {
            long curr = heap.top();
            heap.pop();
            ugly = (int)curr;
            for (int factor : factors) {
                long next = curr * factor;
                if (!seen.count(next)) {
                    seen.insert(next);
                    heap.push(next);
                }
            }
        }
        return ugly;
    }
};
```

```cpp
// priority_queue 优先级队列定义：
priority_queue<Type, Container, Functional>
// Type 就是数据类型，Container 就是容器类型（Container必须是用数组实现的容器，比如vector,deque等等，但不能用 list。STL里面默认用的是vector），Functional 就是比较的方式，当需要用自定义的数据类型时才需要传入这三个参数，使用基本数据类型时，只需要传入数据类型，默认是大顶堆

//升序队列
priority_queue <int,vector<int>,greater<int> > q;
//降序队列
priority_queue <int,vector<int>,less<int> >q;
//greater和less是std实现的两个仿函数（就是使一个类的使用看上去像一个函数。其实现就是类中实现一个operator()，这个类就有了类似函数的行为，就是一个仿函数类了）


//使用例 ：
    //对于基础类型 默认是大顶堆
    priority_queue<int> a;
    //等同于 priority_queue<int, vector<int>, less<int> > a;
    priority_queue<int, vector<int>, greater<int> > c;  //这样就是小顶堆

```

方法二 动态规划

挺难的，不是动态规划方法难，而是题目的思考难。