

给定一个整数 n , 返回从 1 到 n 的字典顺序。

例如,

给定 $n = 13$, 返回 $[1, 10, 11, 12, 13, 2, 3, 4, 5, 6, 7, 8, 9]$ 。

请尽可能的优化算法的时间复杂度和空间复杂度。输入的数据 n 小于等于 5,000,000。

字典排序的话, 因为都是连续的数字, 那么进行深度遍历就可以了, 把上一位的数字乘以 10 加上当前位置。

```
class Solution {
public:
    void dfs(int num, int n, vector<int> &result){
        for(int i=0; i<10; i++){
            if(num*10+i<=n){
                result.push_back(num*10+i);
                dfs(num*10+i, n, result);
            }else{
                break;
            }
        }
    }
    vector<int> lexicalOrder(int n) {
        vector<int> result;
        for(int i=1; i<10; i++){
            if(i<=n){
                result.push_back(i);
                dfs(i, n, result);
            }else{
                break;
            }
        }
        return result;
    }
};
```