0233. 数字 1 的个数

ITCharge 大约 3 分钟

- 标签：递归、数学、动态规划
- 难度：困难

题目链接

- 0233. 数字 1 的个数 - 力扣

题目大意

描述：给定一个整数 n。

要求：计算所有小于等于 n 的非负整数中数字 1 出现的个数。

说明：

- $0 \leq n \leq 10^9$。

示例：

- 示例 1：

```py
输入：n = 13
输出：6
```

- 示例 2：

```py
输入：n = 0
输出：0
```

解题思路

思路 1：动态规划 + 数位 DP

将 n 转换为字符串 s，定义递归函数 `def dfs(pos, cnt, isLimit):` 表示构造第 pos 位及之后所有数位中数字 1 出现的个数。接下来按照如下步骤进行递归。

1. 从 `dfs(0, 0, True)` 开始递归。`dfs(0, 0, True)` 表示：
    1. 从位置 0 开始构造。
    2. 初始数字 1 出现的个数为 0。
    3. 开始时受到数字 n 对应最高位数位的约束。
2. 如果遇到 $pos == len(s)$，表示到达数位末尾，此时：返回数字 1 出现的个数 cnt。
3. 如果 $pos \neq len(s)$，则定义方案数 ans，令其等于 0，即：`ans = 0`。
4. 如果遇到 $isNum == False$，说明之前位数没有填写数字，当前位可以跳过，这种情况下方案数等于 $pos + 1$ 位置上没有受到 pos 位的约束，并且之前没有填写数字时的方案数，即：`ans = dfs(i + 1, state, False, False)`。
5. 如果 $isNum == True$，则当前位必须填写一个数字。此时：
    1. 因为不需要考虑前导 0 所以当前位所能选择的最小数字（$minX$）为 0。
    2. 根据 $isLimit$ 来决定填当前位数字所能选择的最大数字（$maxX$）。
    3. 然后根据 $[minX, maxX]$ 来枚举能够填入的数字 d。
    4. 方案数累加上当前位选择 d 之后的方案数，即：`ans += dfs(pos + 1, cnt + (d == 1), isLimit and d == maxX)`。
        1. `cnt + (d == 1)` 表示之前数字 1 出现的个数加上当前位为数字 1 的个数。
        2. `isLimit and d == maxX` 表示 $pos + 1$ 位受到之前位 pos 位限制。
6. 最后的方案数为 `dfs(0, 0, True)`，将其返回即可。

思路 1：代码

```py
class Solution:
    def countDigitOne(self, n: int) -> int:
        # 将 n 转换为字符串 s
        s = str(n)

        @cache
        # pos: 第 pos 个数位
        # cnt: 之前数字 1 出现的个数。
```

```
        # isLimit: 表示是否受到选择限制。如果为真，则第 pos 位填入数字最多为 s[pos]；如果为假，则最大可为 9。
        def dfs(pos, cnt, isLimit):
            if pos == len(s):
                return cnt

            ans = 0
            # 不需要考虑前导 0，则最小可选择数字为 0
            minX = 0
            # 如果受到选择限制，则最大可选择数字为 s[pos]，否则最大可选择数字为 9。
            maxX = int(s[pos]) if isLimit else 9

            # 枚举可选择的数字
            for d in range(minX, maxX + 1):
                ans += dfs(pos + 1, cnt + (d == 1), isLimit and d == maxX)
            return ans

        return dfs(0, 0, True)
```

思路 1：复杂度分析

- **时间复杂度**：$O(\log n)$。
- **空间复杂度**：$O(\log n)$。