日期模拟

日期模拟的任务是: 在给定起始日期和结束日期的范围内, 统计满足特定条件的日期数量。

处理日期模拟题时,遵循以下步骤进行模拟,通常能够有效避免错误:

- 1. 按天枚举日期:逐天遍历起始日期到结束日期范围内的每个日期。
- 2. 处理闰年:正确判断闰年条件。闰年定义为:年份 x 满足以下任意一个条件:
 - 是 400 的倍数。
 - 是4的倍数但不是100的倍数。

艺术与篮球

C++

```
#include<bits/stdc++.h>
   using namespace std;
2
   // 每月天数数组,平年默认2月28天
   int months[] = \{0, 31, 28, 31, 30, 31, 30, 31, 30, 31, 30, 31\};
   // 对应数字的权值数组
   int hz[] = \{13, 1, 2, 3, 5, 4, 4, 2, 2, 2\};
7
9
   // 判断闰年函数
10
   int leap(int year) {
       // 闰年条件: 是400的倍数,或者是4的倍数但不是100的倍数
11
12
       return (year % 400 == 0) || (year % 4 == 0 && year % 100 != 0);
13
   }
14
15
   // 主逻辑处理函数
   void solve() {
16
       int ans = 0; // 符合条件的日期计数
17
       for (int year = 2000; year <= 2024; year++) {
18
19
           // 设置当前年份的2月天数
20
           if (leap(year)) months[2] = 29;
21
           else months[2] = 28;
22
           // 遍历每月
23
           for (int month = 1; month <= 12; month++) {</pre>
24
               // 遍历每天
               for (int day = 1; day <= months[month]; day++) {</pre>
25
                   int cnt = 0; // 当前日期的权值总和
26
27
                   int y1, y2, y3, y4, m1, m2, d1, d2;
28
                   // 提取年份、月份、日期的各个位
29
                   y1 = year / 1000;
30
                   y2 = year / 100 \% 10;
31
                   y3 = year / 10 \% 10;
32
                   y4 = year \% 10;
33
                   m1 = month / 10;
34
                   m2 = month \% 10;
35
                   d1 = day / 10;
                   d2 = day \% 10;
36
37
                   // 计算权值总和
38
                   cnt = cnt + hz[y1] + hz[y2] + hz[y3] + hz[y4]
                         + hz[m1] + hz[m2] + hz[d1] + hz[d2];
39
```

```
40
                   // 若权值总和大于50, 计入答案
41
                   if (cnt > 50) ans++;
42
43
                   // 若到达目标日期2024年4月13日,输出结果并结束
44
                   if (year == 2024 && month == 4 && day == 13) {
45
                       printf("%d", ans);
46
                       return;
47
                   }
48
               }
49
           }
       }
50
51
   }
52
   int main() { // 日期范围: 2000-01-01到2024-04-13
53
        solve();
   }
54
```

Java

```
public class Main {
 1
 2
        // 每月天数数组,平年默认2月28天
 3
        static int[] months = {0, 31, 28, 31, 30, 31, 30, 31, 30, 31, 30,
    31};
        // 对应数字的权值数组
 4
 5
        static int[] hz = \{13, 1, 2, 3, 5, 4, 4, 2, 2, 2\};
 6
        // 判断闰年函数
8
        static boolean isLeapYear(int year) {
 9
            // 闰年条件: 是400的倍数,或者是4的倍数但不是100的倍数
            return (year % 400 == 0) || (year % 4 == 0 && year % 100 != 0);
10
11
        }
12
13
        // 主逻辑处理函数
        public static void solve() {
14
           int ans = 0; // 符合条件的日期计数
15
16
            for (int year = 2000; year <= 2024; year++) {
17
                // 设置当前年份的2月天数
18
               months[2] = isLeapYear(year) ? 29 : 28;
19
20
               // 遍历每月
21
               for (int month = 1; month <= 12; month++) {</pre>
22
                   // 遍历每天
23
                   for (int day = 1; day <= months[month]; day++) {</pre>
24
                       int cnt = 0; // 当前日期的权值总和
25
                       int y1, y2, y3, y4, m1, m2, d1, d2;
26
27
                       // 提取年份、月份、日期的各个位
28
                       y1 = year / 1000;
29
                       y2 = year / 100 \% 10;
30
                       y3 = year / 10 \% 10;
31
                       y4 = year \% 10;
32
                       m1 = month / 10;
33
                       m2 = month \% 10;
34
                       d1 = day / 10;
35
                       d2 = day \% 10;
36
37
                       // 计算权值总和
38
                       cnt += hz[y1] + hz[y2] + hz[y3] + hz[y4]
```

```
39
                             + hz[m1] + hz[m2] + hz[d1] + hz[d2];
40
41
                        // 若权值总和大于50, 计入答案
42
                        if (cnt > 50) ans++;
43
44
                        // 若到达目标日期2024年4月13日,输出结果并结束
45
                        if (year == 2024 && month == 4 && day == 13) {
46
                            System.out.println(ans);
47
                            return;
48
                        }
49
                   }
50
               }
51
            }
52
        }
53
        public static void main(String[] args) { // 日期范围: 2000-01-01到2024-
54
    04-13
55
            solve();
56
        }
    }
57
58
```

Python

```
# 每月天数数组,平年默认2月28天
 2
   months = [0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
 3
   # 对应数字的权值数组
   hz = [13, 1, 2, 3, 5, 4, 4, 2, 2, 2]
 4
 5
   # 判断闰年函数
 6
 7
    def is_leap_year(year):
8
       # 闰年条件: 是400的倍数,或者是4的倍数但不是100的倍数
9
       return year % 400 == 0 or (year % 4 == 0 and year % 100 != 0)
10
   # 主逻辑处理函数
11
12
    def solve():
13
       ans = 0 # 符合条件的日期计数
14
       for year in range(2000, 2025):
15
           # 设置当前年份的2月天数
16
           months[2] = 29 if is_leap_year(year) else 28
17
18
           # 遍历每月
19
           for month in range(1, 13):
20
               # 遍历每天
21
               for day in range(1, months[month] + 1):
22
                   cnt = 0 # 当前日期的权值总和
23
24
                   # 提取年份、月份、日期的各个位
25
                   y1, y2, y3, y4 = year // 1000, year // 100 % 10, year // 10
   % 10, year % 10
26
                   m1, m2 = month // 10, month % 10
                   d1, d2 = day // 10, day \% 10
27
28
29
                   # 计算权值总和
30
                   cnt += hz[y1] + hz[y2] + hz[y3] + hz[y4] \setminus
31
                         + hz[m1] + hz[m2] + hz[d1] + hz[d2]
32
```

```
33
                 # 若权值总和大于50, 计入答案
34
                 if cnt > 50:
35
                     ans += 1
36
37
                 # 若到达目标日期2024年4月13日,输出结果并结束
38
                 if year == 2024 and month == 4 and day == 13:
39
                     print(ans)
40
                     return
41
42 # 日期范围: 2000-01-01到2024-04-13
43 solve()
44
```