NOIp Senior Day2 Solution

zhzh2001

1 围栏

1.1 思路

一个正方形可以由左上角坐标和边长确定，只要枚举所有正方形就能找到答案。这样的时间复杂度 = 枚举左上角坐标的时间 * 枚举边长的时间 * 统计正方形内的草场的时间。直接枚举 + 离散化是 $O(N^4)$ 的。我们可以做二维前缀和，并二分枚举边长，并用单调性在 $O(N^2)$ 的时间内判断，总的复杂度是 $O(N^2 \log N)$ 的。

1.2 常数问题

由于 $N \leq 3,000$，坐标范围 $0 \ldots 10^{18}$，上述方法较难直接通过。而官方的实现常数小，也利用了单调性，但是时间复杂度难以分析，平均情况下比二分慢，但是在较坏情况下比二分快，而且与坐标范围无关。

1.3 总结

本题来自`USACO06JAN Gold T3:Corral the Cows`
主要考察了二分和单调性。

2 Quicksort Killer

2.1 直接排序

在我们印象中，快速排序在正序或逆序时会达到 $\Theta(N^2)$。因此直接对 N 个数排序输出。这样就有 50 分了，额外的 10 分来自于随机种子等于 0 的情况，此时和第二种实现完全一样。

2.2 思路

要使快速排序达到最坏情况，只要让基准值是当前序列中最小的即可 (最大也可以)。分析可以发现，这样只会扫描一遍，并把基准值与第一个数交换，$T(N) = T(N-1) + \Theta(N) = \Theta(N^2)$。具体实现维护一个数组 B 记录原先的位置，每次只要在基准值填上当前最小数，并交换 $B[l]$ 和 $B[F[l,r]]$。

2.3 时间计算

这也是比较容易出错的部分，可以直接枚举，也可以推出公式[1]：

```
tm_sec + tm_min*60 + tm_hour*3600 + tm_yday*86400 +
    (tm_year-70)*31536000 + ((tm_year-69)/4)*86400 -
    ((tm_year-1)/100)*86400 + ((tm_year+299)/400)*86400
```

其中`tm_yday` 表示从 1 月 1 日到该时刻经过的天数。

另一个问题是在 Windows 下，计算结果必须减掉 8*3600 秒，因为我们的时区为 UTC+8。这也是我强调 UTC 的原因。

2.4 使用日期函数计算

实际上，C 语言的`time.h` 中有一个`mktime` 函数可以直接完成转换。其原型如下：

```
time_t mktime( struct tm *time );
```

而上述公式中的值都是结构体`tm` 中的成员，另外还有`tm_mon` 和`tm_mday`，可以替换`tm_yday`。但是请注意这些值的范围。[2]

2.5 总结

本题是我的原创题，考察了对快速排序的理解，以及日期模拟。

3 花园改造

3.1 暴力

可以把当前 N 个花坛内的泥土状压或 hash，然后就转化为最多 C^N 个点的最短路。(设 $A_i, B_i \leq C$)

[1]参见`http://pubs.opengroup.org/onlinepubs/9699919799/basedefs/V1_chap04.html#tag_04_16`

[2]参见`http://en.cppreference.com/w/c/chrono/tm`

3.2 动态规划

通过一个巧妙的模型转化，就可以把这题转化为一个字符串编辑距离问题。只要按顺序把每个花坛 i 内的泥土数量展开成对应数量的 i，如 1,2,3,4 转化为 1,2,2,3,3,3,4,4,4,4。接下来，购买操作就可以转化为插入一个对应的数字 (花费 X)，移除操作转化为删除一个对应的数字 (花费 Y)，运送操作转化为修改一个对应的数字 (花费 $Z*|i-j|$)。这样时间复杂度 $O((NC)^2)$。

3.3 贪心

官方题解时间复杂度为 $O(NC)$，但是分析很复杂，有兴趣可以去研究。

这里介绍一种 CF 上有人提出的贪心做法。如果 $A_i > B_i$，对于每个单位多余的泥土，可以花费 Y 把它移除，或者运送到前面的花坛 j 花费 $Z*(i-j)$(这里强制 $i > j$，$i < j$ 的情况会在 $A_i < B_i$ 时考虑)。我们如何知道哪种更优呢？$Z*(i-j) = Z*i - Z*j$，只要找到最小的 $-Z*j$ 就可以了。当然我们还要考虑退钱，因为可能在一开始直接移除更优，而后来发现运送更优，这时就要退钱。

一般的，我们维护两个小根堆，分别维护多余和缺少的泥土的 $-Z*i-cost$，其中 $cost$ 为原费用，初始时为 Y 和 X。如果 $A_i > B_i$，$cost = min(Y, Q_{need}.top + Z*i)$，然后向多余堆插入 $-Z*i-cost$；$A_i < B_i$ 同理。时间复杂度 $O(NC \log NC)$，空间复杂度 $O(NC)$。