云斗 CSP-J 模拟赛题解

1　不平凡的数

$x^2 - y^2 = (x+y)(x-y)$。而 $x+y$ 和 $x-y$ 的奇偶性相同，因此合法的必要条件是 N 是奇数或 4 的倍数。

事实上这是充要的，我们给出构造:

- 对于奇数 N，令 $x = (N+1)/2$，$y = (N-1)/2$；
- 对于偶数 N，令 $x = N/4+1$，$y = N/4-1$。

2　文物修复

做法一

发现 n 等于 10，所以直接 dfs 爆搜每天工具校准还是工具保养。

时间复杂度 $O(2^n)$。

期望得分 10 分。

做法二

假设校准的天数固定了，那么我们可以考虑让前面的天数都是校准天数，因为这样可以让我们得到的一定更有利于我们完成相应的文物修复。所以我们可以直接去枚举校准的天数，假设前 k 天都在工具修复，那么就会得到

$$x, 2x, 3x, \ldots, kx, (k-1)x, \ldots, 2x, x$$

一共 $2k-1$ 个熟练度。然后把这些熟练度从大往小排序之后，用双指针去贪心匹配文物难度。

时间复杂度 $O(n^2)$。

期望得分 60 分。

做法三

延续做法二，我们可以发现，校准的天数具有单调性，前面几天，校准的天数越多，越有利于完成全部的文物修复，所以我们可以二分校准的天数，然后 check 的时候，使用做法二的方法即可。

时间复杂度 $O(n \log n)$。

期望得分 100 分。

3 抽奖

一个长为 n 的序列 a_i，有 m 次询问。每次询问一个区间 $[l, r]$，在 $[l, r]$ 中随机选取两个数（两次选取互不干扰）x, y。求

$$\sum_{i=\min(x,y)}^{\max(x,y)} a_i$$

的和值。

首先，两次抽取互不干扰即一共有 $(r-l+1)^2$ 种情况。

问题就变为了

$$\sum_{i=l}^{r} \sum_{j=l}^{r} \sum_{k=\min(i,j)}^{\max(i,j)} a_k$$

最暴力的方法就是直接求上面的式子。复杂度 $\Theta(n^4)$。可以获得 $0pts$。

考虑优化，最后一个 $\sum$ 显然可以用前缀和优化掉。变为求

$$\sum_{i=l}^{r} \sum_{j=l}^{r} (sum_{\max(i,j)} - sum_{\min(i,j)-1})$$

复杂度 $\Theta(n^3)$，可以获得 $20pts$。

考虑进一步优化，发现仍可以用前缀和进行优化。式子有些复杂在此不多赘述。

这样一步步用前缀和优化下去，复杂度 $\Theta(n^2)$ 或 $\Theta(n)$。可以获得 $40 \sim 100pts$。

对于 $1 \le n \le 10^5$，显然可以莫队维护。（不过应该没有人会写这种方法吧。）

接下来介绍另外一种维护方式，考虑每个数的贡献。

第 i 个数的贡献为

$$[2 \times (i-l+1) \times (r-i+1) - 1] \times a_i$$

将上式展开可以得到

$$[2 \times (-i^2 + (l+r) \times i + r - l + 1 - lr) - 1] \times a_i$$

那么就只用分别维护 $a_i, a_i \times i, a_i \times i^2$ 的前缀和代入上式即可。复杂度 $\Theta(n)$。

4 芳香串

考虑枚举选出来的"1", "4", "5"分别是哪几个字符，依次进行动态规划。

设 $dp_{i,0}$ 表示考虑到第 i 个字符且第 i 个字符必须选择，形如"1"的串有几个。

同理，设 $dp_{i,1}$ 表示形如"11"的串，$dp_{i,2}$ 表示形如"114"的串，$\cdots$，$dp_{i,5}$ 表示形如"114514"的串有几个。

转移方程大致如下（注意下面的"1"、"4"、"5"并不是真正的数字 $1, 4, 5$，而是枚举出来的三个字符）：

$$dp_{i,1} = [t_i = \text{'1'}] \sum_{j=1}^{i-1} dp_{j,0}$$

$$dp_{i,2} = [t_i = \text{'4'}] \sum_{j=1}^{i-1} dp_{j,1}$$

$$dp_{i,3} = [t_i = \text{'5'}] \sum_{j=1}^{i-1} dp_{j,2}$$

$$\vdots$$

最终答案为：

$$\sum_{i=1}^{n} dp_{i,5}$$

上述暴力转移的时间复杂度为 $O(n^2m^3)$，可以使用前缀和优化到 $O(nm^3)$，但仍然无法通过。

进一步优化时，可以用 `vector` 记录 m 个字符各自出现的位置，仅在 t_i 等于上面三个字符之一时进行转移。这样，每个字符只会被枚举 m^2 次，复杂度降为 $O(nm^2)$，但依旧不能通过。

最后，我们可以只枚举选出来的"1"与"4"两个字符。对于 dp_2 到 dp_4 的转移，只需考虑中间有多少个字符不等于"1"和"4"即可。结合上述优化，可将复杂度降至 $O(nm)$，从而顺利通过所有测试点。