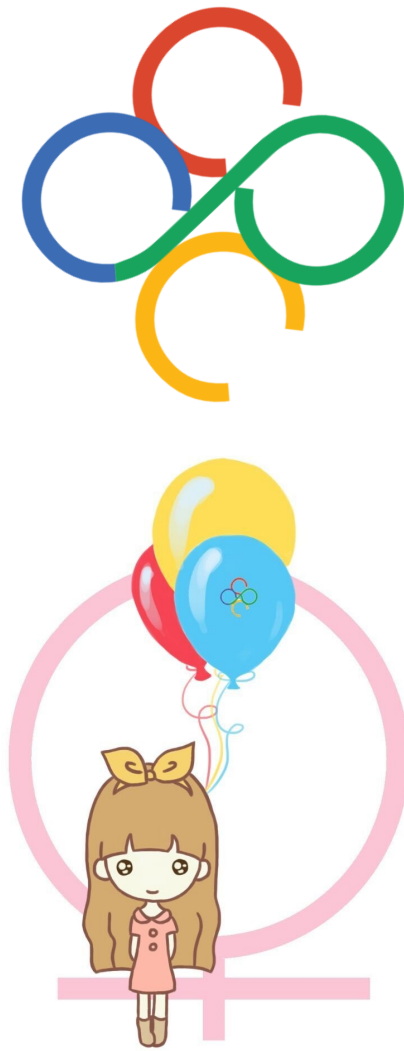


2021 年中国大学生程序设计竞赛 女生专场 - 赛题分析



山东, 淄博

2021 年 10 月 31 日

1 公交线路

Shortest judge solution: 448 Bytes.

检查坐对方向和坐反方向两种情况对应的报站序列是否符合输入，如果都符合那就是“Unsure”，否则可以确定有没有坐反。

2 攻防演练

Shortest judge solution: 720 Bytes.

考虑如何判断一个字符串 $t = t_1 t_2 \dots t_k$ 是否是 $s_{l,r} = s_l s_{l+1} \dots s_r$ 的子序列：用一个指针从 l 开始往右进行扫描，找到第一个 t_1 并停下来，然后从那个位置接着往右找到第一个 t_2 ，... 如果按这个贪心方式能找完所有 k 个字符，那么 t 就是 $s_{l,r}$ 的子序列；如果指针扫到了 $r+1$ 甚至更远的地方，那么说明 t 不是 $s_{l,r}$ 的子序列。

现在考虑如何寻找一个最短的 t ，使得 t 不是 $s_{l,r}$ 的子序列。假设指针目前位于 x ， t 的下一个字符有 m 种选项，选第 i 种选项时，指针将扫到 x 右边第一个字符 i 的位置 $v_{x,i}$ （如果不存在 i 那么 x 将扫到 $n+1$ ）。那么为了使得指针往右扫描得尽量远， t 的下一个字符应该选择 $v_{x,i}$ 最大的那个 i 。因此，预处理出 $next_x$ 表示指针位于 x 时，下一步将扫到哪个位置，无论询问的 $[l,r]$ 是什么，它都是一个定值，不随 $[l,r]$ 而改变。

于是问题转化为：从 l 开始沿着 $next$ 一路往右跳，要跳多少步才能跳到 $> r$ 的地方？这是个经典问题，可以使用倍增；也可以离线询问后，将 $next_x$ 当作 x 的父亲建出一棵有根树，在树上进行二分查找。

时间复杂度 $O(nm + q \log n)$ 。

3 连锁商店

Shortest judge solution: 904 Bytes.

如果某家公司开的连锁店数量不超过 1，那么可以无视“每家公司的红包只能领一份”这个限制，这是因为任何一条路线都无法访问多次该公司开的商店。如果某家公司开的连锁店数量至少为 2，那么这样的公司数最多为 $\frac{n}{2} \leq 18$ 。

由于第二类公司数量并不多，因此可以使用状态压缩动态规划来求解这个问题。设 $f[i][S]$ 表示从 1 出发到达了 i 点，一路上访问过的第二类公司集合为 S 时，访问过的第一类公司的红包总价值最大是多少，枚举下一个景点进行转移。

时间复杂度 $O(n^2 2^{\frac{n}{2}})$ 。

4 修建道路

Shortest judge solution: 190 Bytes.

不妨设整个序列的最大值位于 a_x （若有多个最大值则取最小的 x ），那么两端点都在 $[1, x-1]$ 或都在 $[x+1, n]$ 的边的边权不会超过 a_x ，而横跨 x 的边的边权都是 a_x ，因此最优方案一定是左右两部分都分别连成一个连通块后，再和 x 相连。

于是对于每个 x 计算它需要连多少条边：如果 $a_{x-1} < a_x$ ，那么左边对应的连通块要和 x 连一条边；如果 $a_{x+1} \leq a_x$ ，那么右边对应的连通块要和 x 连一条边。因此：

$$ans = \sum_{i=1}^{n-1} \max(a_i, a_{i+1})$$

5 被遗忘的计划

Shortest judge solution: 809 Bytes.

令每件商品的价值最大值为 t ，那么选取 k 件商品的价值最大值不超过 $k \times t$ ；另一方面，我们总可以把 t 对应的商品重复选取 k 次来得到 $k \times t$ ，因此数组 f 的最大值一定对应 $k \times t$ ，于是我们得到了 k 的唯一可能取值：即两个数组最大值的商。

得到 k 的唯一可能取值后，使用快速幂求出 v 数组的循环卷积的 k 次幂，判断是否等于 f 数组即可。

时间复杂度 $O(n^2 \log k)$ 。

6 地图压缩

Shortest judge solution: 878 Bytes.

不难发现行与列是两个独立的问题，因此只需要求出行的最短循环节的长度，再求出列的最短循环节的长度，相乘就是答案。

以行为例，首先通过 Hash 将问题转化为一维问题。一维问题则是经典问题，对于一个长度为 n 的字符串，长度为 d 的前缀是循环节当且仅当长度为 $n - d$ 的后缀相等，因此需要找到这个字符串最长的前缀，满足该前缀也是该字符串的后缀。可以枚举所有可能的 d 然后使用 Hash $O(1)$ 判断；也可以使用 KMP 算法求出 $next$ 数组，答案即为 $n - next[n]$ 。

时间复杂度 $O(n^2 + qn)$ 。

7 3G 网络

Shortest judge solution: 82 Bytes.

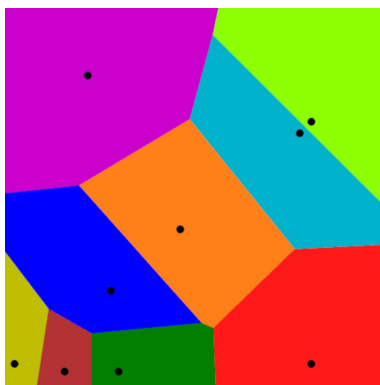
当 $r \rightarrow +\infty$ 时， n 个圆的并趋近于一个半径为 r 的圆，因此答案为 $\frac{1}{n}$ 。

8 4G 网络

Shortest judge solution: 3620 Bytes.

对于每个询问，我们要求的是这 n 个半径为 r 的圆的面积之并（除以它们的总面积 $n\pi r^2$ ）。

根据微积分的概念，一个区域的面积等价于区域中每个微元累加的结果。注意到所有圆的半径都相等，对于平面中的每个点，它属于圆并当且仅当存在一个圆的圆心到它的距离不超过 r 。因此对于每个点，我们将其放在离它最近的圆心处考虑，如果离它最近的圆心到它的距离不超过 r ，那么它需要被计入答案。



枚举每个圆心，找到在 $[-inf, inf] \times [-inf, inf]$ 这个矩形里它作为最近点的管辖区域，容易发现是一个凸多边形，如上图所示。枚举每个圆心，再枚举另一个圆心，可行区域是它们的垂直平分线的一侧，可以使用半平面交在 $O(n^2 \log n)$ 时间内预处理出所有 n 个凸多边形，即 Voronoi 图。根据 Voronoi 图的性质，所有凸多边形的边数之和为 $O(n)$ 。

对于每个询问，枚举一个圆心 O ，再枚举它管辖区域的凸多边形的一条边 (A, B) ，那么对答案的贡献为三角形 OAB 与圆 O 的交，可以 $O(1)$ 计算得到。

时间复杂度 $O(n^2 \log n + qn)$ 。

9 驾驶卡丁车

Shortest judge solution: 848 Bytes.

按照题意逐指令逐步模拟即可。

10 最大权边独立集

Shortest judge solution: 1197 Bytes.

枚举位于最终边独立集上的加入的边权为 p 的边的数量 t ，那么 $0 \leq t \leq k$ 且 $2t \leq n$ ，这是因为每条边将占据图中的两个点。

假设最终要加入 t 条边，那么需要从图中删去 $2t$ 个点，然后用 $t \times p +$ 剩下图的最大权边独立集来更新答案，这等价于在树上规定 $2t$ 个点不匹配其它点，然后计算树的带权最大匹配。

使用自底向上的树形动态规划来解决这个问题：设 $f[i][j][0]$ 表示考虑了 i 点的子树， i 点的子树内删掉了 j 个点，且 i 不能往上匹配 i 的父亲时的带权最大匹配；设 $f[i][j][1]$ 表示考虑了 i 点的子树， i 点的子树内删掉了 j 个点，且 i 能够往上匹配 i 的父亲时的带权最大匹配。那么状态数为 $O(nk)$ ，在转移时需要合并两棵子树的信息， j 这一维从 0 开始枚举到 $\min(size_x, k)$ 即可保证时间复杂度为 $O(nk)$ ，其中 $size_x$ 表示 x 目前的子树大小。

11 音乐游戏

Shortest judge solution: 132 Bytes.

根据输入数据的合法性，问题等价于统计输入的所有字符串中“-”的个数。一个简单的实现方式是：`while(~scanf("%s",s))` 统计 s 中“-”的个数。