# Solution

## Summary

- 出题人自认为按照「不超过区域赛 Ag 水平」约定严格控制难度，有的题就是能加强但没加强就嗯让大家过。
- 结果随便翻了下前几场，感觉这一场好像成十场里比较简单的了。产生自我怀疑了，到底是谁的问题，严查。
- （虽然也有的题是因为写 STD 的时候想复杂了所以验完题发现比预计简单？）
- 总之祝大家把握机会 AK！
- Update：为 1002 和 1005 的 Notification 致歉。Preparation 难免有疏漏之处，望大家谅解。

## 1001 公交通勤

- 易得：
    - 落入长度为 $d$ 的等车区间的概率为 $\frac{d}{nt}$，此时期望等车时间为 $\frac{d}{2}$。
    - 因此，全局期望等待时间为 $\sum \frac{d^2}{2nt}$。
    - 因此，原题意等价于最小化 $\sum d^2$。
- 令 $dp_{cur,i,j}$ 表示前 $i-1$ 趟车已过站、在第 $cur$ 分钟时第 $i$ 趟车到达、且当前共应用 $j$ 次操作时的最小期望。
    - 转移时，考虑枚举上一趟车的到达时间为 $lst$，有

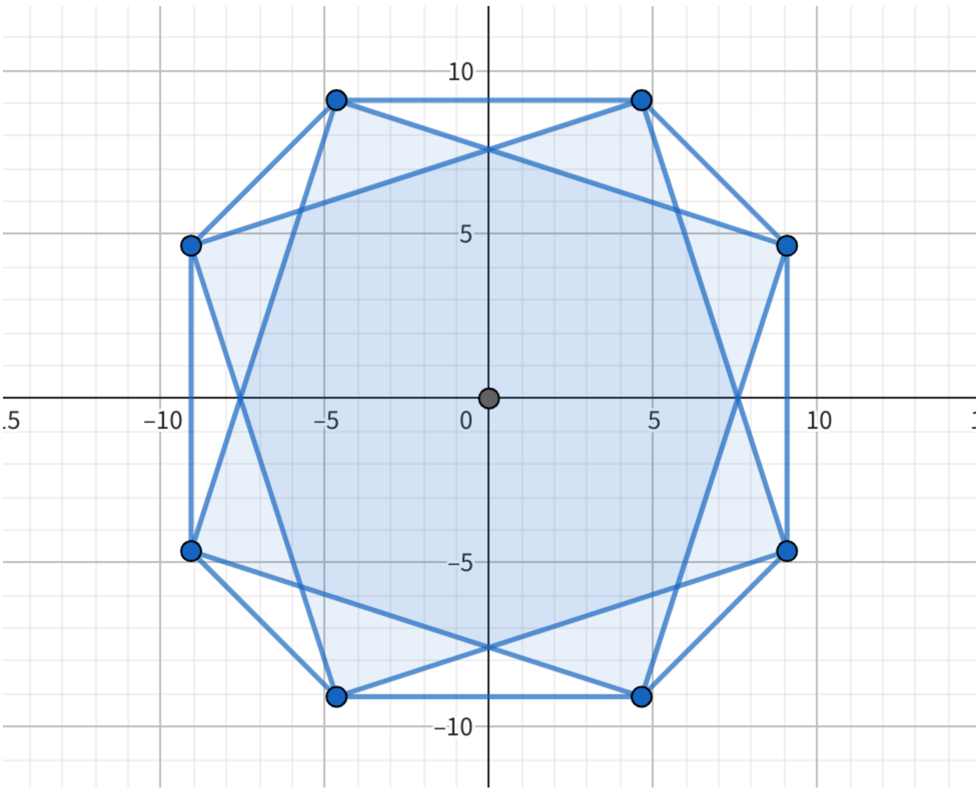
$$dp_{cur,i,j} \leftarrow dp_{lst,i-1,j-cnt} + (cur - lst)^2$$

    ，其中 $cnt = |a_i - cur|$。
    - 转移到最后一趟车时，需注意加上 $d = nt - a_{n-1}$ 时段的贡献。
    - 最终答案为 $\min_{cur,j}(dp_{cur,n-1,j})$。
- 朴素 DP 时间复杂度为单组 $O(nm(nt)^2)$，实际常数较小，可以通过本题。

# 1002 折线绘制

- STD 做法：
    - 中心对称的区间有中心扩展性，即：
        - 如果 $[mid-r, mid+r]$ 中心对称，
        - 且 $a_{mid-r-1}+a_{mid+r+1}=a_{mid-r}+a_{mid+r}$，
        - 则 $[mid-r-1, mid+r+1]$ 中心对称。
        - （应用 Manacher 的插空法处理奇偶性时，以上 $\pm 1$ 相应为 $\pm 2$）
    - 因此，我们可以使用 Manacher 算法处理此类中心扩展问题。
    - 时间复杂度为单组 $O(n)$。
- 其他做法：Hash
    - 有不止一个验题人在写 Hash。
    - 但是实现起来比 Manacher 费劲很多。
    - 而且有 Hash 冲突 WA / 写得丑大常数 TLE 的风险。

# 1003 乌蒙大象




- 不难发现，一个乌蒙可以拆成两个大小相等、共用中心的正方形。
- 考虑枚举正方形。
    - 对于正方形 $ABCD$，枚举 $A, B$ 之后 $C, D$ 就确定了。
    - 因此，使用 set 记录所有点即可 $O(n^2 \log n)$ 枚举所有正方形。
- 同时使用 map 套 map 统计每个中心点、每种边长的正方形数量，即可计算答案。
    - $k$ 个中心点为 $(x_0, y_0)$ 边长为 $L$ 的正方形会对答案产生 $\binom{k}{2}$ 的贡献。
    - 因为从这 $k$ 个正方形中任选两个都能构成乌蒙。

- 时间复杂度为单组 $O(n^2\log n)$。
- 需要注意实现的常数和细节。如果写得丑，就会 TLE。

# 1004 储值购物

- 容易发现这样的性质：
  - 在一张卡上一次性消费 $A$ 金额，或在一张卡上两次分别消费 $B+C=A$ 金额，前者得到的答案总是不劣于后者。
  - 因此，在结束时刻，所有余额为 $x$ 的卡都应该被视为一次性消费 $W-x$ 金额而得到。
  - 因此，在结束时刻，不可能存在两张卡的余额 $x_u+x_v>W$。否则这两次 $W-x_u$ 和 $W-x_v$ 金额的消费可被同一张卡完成。
- 由于以上性质，在结束时刻从小到大排列余额 $x_1,x_2,\ldots,x_{ans}$，一例最优方案将形如 $x_1\le x_2=x_3=x_4=\cdots=x_{ans-1}\le x_{ans}$。
  - $W+1$ 为偶数时，$x_1=x_2=\frac{W+1}{2}$；
  - $W+1$ 为奇数时，$x_1=\left\lfloor\frac{W+1}{2}\right\rfloor, x_2=\left\lceil\frac{W+1}{2}\right\rceil$；
  - $x_{ans-1}$ 与 $x_{ans}$ 未必相等的原因：开出前 $ans$ 张卡后可能剩余少量预算但无法开出第 $ans+1$ 张卡，此时花在哪张卡上并无实质区别。
- 因此，令 $x_2=\left\lceil\frac{W+1}{2}\right\rceil$，则答案为 $\left\lfloor\frac{V+[W \bmod 2=0]}{k}\right\rfloor$。
- 注意至少需要一张卡支付，因此答案仍需和 $1$ 取 max。
  - 第一版的 STD 和前三个验题人同时忘记了这个问题...
  - 所以必须特别感谢 frost_ice，不然就要出锅了。
  - 本来想了一下要不要不给第三个样例，但是为了控制难度忍住了。

# 1005 真爱口上

- STD 做法：
  - 注意到以下性质：
    - 基本规则中，`a/i/u/e/o` 总是对应 1 Mora；
    - 额外规则中，`p/t/k/s/n` 的双写也总是对应 1 Mora；
    - 容易验证没有其他的 Mora。
  - 唯一的要点在于 `...nn + n...` 的情况中不要把三连 `n` 统计为两次双写 `n`。
  - 因此，容易在线性时间复杂度内统计 Mora 数量。
- 也可以有很多另外的同样线性但并不那么好写的做法。
- 由于正解不依赖表格的具体内容，因此发现表格错误后没有第一时间考虑发 Notification，再次致歉。

# 1006 三角障碍

- 考虑以下转化：
  - 将 $n$ 个障碍物、$3$ 条边 $AB, BC, CA$ 视为 $n+3$ 个结点；
  - 将障碍物/三角形边的几何相交对应为图上的连边。
- 则原题意转化为选择最小代价点集使结点 $AB, BC, CA$ 两两可以通过选中结点连接。
  - 这样的连接对应于原题意中通过连续相交的障碍物到达另一端。
- 为求解新题意，我们可以使用 Steiner 树。
- 但是，注意到只需求解 $k=3$ 的 Steiner 树，故以下做法可以得到同样优的答案：
  - 分别从结点 $AB, BC, CA$ 出发，做三次 Dijkstra。
    - 此处需应用点权转边权的技巧，将当前点的点权作为向前走一步的边权。
  - 枚举中心点 $u$ 求 $\min_u(dis_{AB,u} + dis_{BC,u} + dis_{CA,u} + w_u)$ 即得答案。
- 视 Dijkstra 优化程度，单组时间复杂度为 $O(n^2)$ 或 $O(n^2 \log n)$。
- 注意仔细处理判断结点相交的条件。
  - 为此 STD 写了一些奇奇怪怪的处理…
  - 一个验题人提出的简单方案是两障碍物交 iff 至少有一个顶点在另一个三角形内。

# 1007 扑克洗牌

- 注意到：
  - 如果在值域上存在一段连续段 $l, l+1, \dots, r$，
  - 满足 $a_l < a_{l+1} < \dots < a_r$，
  - 则操作次数可以为 $n-(r-l+1)$。
- 原因是：
  - 我们可以将值域 $[l, r]$ 内的牌视为始终没有被操作过的，
  - 其他牌都被操作过而可以插入到结果中的任意位置。
- 因此，只需找到最长的这样的值域连续段，就可以找到最小的操作次数。
- 考虑记 $pos_{a_i} = i$，题意转化为在 $pos$ 序列上找到最长的单增子段。
- 这容易在线性时间复杂度内完成。

# 1008 饮水计划

- 由数据范围可得，做法几乎只可能是预处理所有区间答案并 $O(1)$ 回答单次询问。
- 以下考虑如何预处理所有区间答案。
- STD 做法：
  - 考虑这样的性质：
    - 对于某个正整数 $x$，记 $b_i = [a_i > x]$。

        - 考虑 $x$ 从小到大增加、$b$ 序列从全 $1$ 逐渐变为全 $0$ 的过程。
        - $[l,r]$ 存在 milestone $k$ iff 某一时刻 $[l,k]$ 全 $0$ 且 $[k+1,r]$ 全 $1$。
    - 我们考虑如何一边模拟这样的渐变，一边不重不漏地计数。
        - 每次将某个 $b_i$ 从 $1$ 变为 $0$ 时，考虑 $i$ 所在的 $0$ 连续段 $[l_0,r_0]$。
        - 再考虑 $[l_0,r_0]$ 右侧的 $1$ 连续段 $[l_1,r_1]$。
        - 一个朴素的想法是，令所有 $x\in[l_0,r_0], y\in[l_1,r_1]$ 的 $ans_{x,y}$ 加 $1$，代表此时从 $x$ 到 $y$ 可以选取 $i$ 作为 milestone。
        - 容易看出，这样计数是有重复的。实际上，一个正确的计数方法是，令所有 $x\in[l_0,i], y\in[l_1,r_1]$ 的 $ans_{x,y}$ 加 $1$。想一想为什么。
    - 因此，原问题转化为二维区间加后静态查询单点值。
        - 使用二维前缀和可以在单组 $O(n^2+q)$ 的时间复杂度内解决该问题。
        - 可以注意到，转化后的题意也可以增强数据范围使用树套树等做法通过。
            - 但是因为控制难度（而且我也不想写 STD）所以算了。
- 其他做法：枚举 $l$ 再枚举 milestone $k$ 再二分 / 双指针 $r$
    - 一开始没想到把树套树范围削成现在的范围之后会跑出来新做法，疏忽了。
        - 实际上验题人基本都在搞这个。估计正赛也会是这样。
        - 比 STD 好想很多导致预计难度显著下降，早知道出树套树跟大家爆了。
    - 时间复杂度视做法分别为单组 $O(n^2\log n+q)$ 或 $O(n^2+q)$。
        - 如果你写的是前者，可能时限会很紧。
        - 想了一下要不要干脆把所有二分的都杀了。
        - 但要去掉 I/O 瓶颈才比较好杀 log（直接询问所有区间答案之和或者什么）。
        - 担心这样询问又被搞出更简单的做法，算了为了控制难度还是放大家过吧。

# 1009 自由落体

- 线段树练习题。
- 正确做法：
    - 考虑从下向上排序所有矩形后依次处理每个矩形。
    - 考虑使用线段树维护每个横坐标上的最高点。对于某个矩形：
        - 查询其底边可以下落到的高度 等价于 线段树上询问区间 max；
        - 模拟该矩形下落到该高度并占据对应位置 等价于 线段树上区间 assign。
        - 在处理每个矩形时顺便计算做功即可。
    - 注意横坐标的一些 $\pm 1$ 的处理。这是基本功所以故意没有在样例体现。
    - 时间复杂度为单组 $O(n\log n+n\log W)$。其中 $W=2\times 10^5$。
- STD 的犯蠢做法：
    - 题意来自实际需求，其中需在「某一竖列经过的所有矩形」序列上维护相邻关系。
    - 所以 STD 用扫描线 + 拓扑排序处理矩形的顺序。看到验题代码之后发现好多余……

# 1010 绳子切割

- 正确做法:
    - 倒过来观察整个过程。
        - 可以发现，如果存在合法的切割顺序，则前 $i$ 个结点应该不通过编号更大的结点而形成一个连通块。
        - 反之，如果第 $i$ 个结点必须经过编号大于 $i$ 的结点才可能连通前 $i-1$ 个结点构成的连通块，则不存在合法的切割顺序。
    - 因此，可以归纳出以下做法:
        - 从 $1$ 至 $n$ 枚举 $i$，如果存在 $j<i$ 满足 $i,j$ 直接相连，则称 $i$ 合法。
        - 最终存在合法切割方案 iff 结点 $1$ 至 $n$ 均合法。
    - 时间复杂度为单组 $O(n)$。
- 可以想象会有五花八门的写法，但它们与正解本质相同，只是实现得更加复杂。
- 作为例子的 STD 做法:
    - 冗余地使用了并查集模拟上述过程:
        - 枚举 $i$，连接结点 $i$ 与更小编号结点的所有边，询问 $i$ 是否与 $0$ 连通。
    - 时间复杂度为单组 $O(n\alpha(n))$。