

Water

算法一：

每个圆柱体新增的体积是 $(h[i]-h[i-1]) * (r[i-1]-r[i])^2$ ，直接计算即可。

复杂度 $O(n)$ ，期望得分 30 分。

算法二：

对圆柱体按照高度从小到大进行排序，转化为算法一。考察 stl 排序算法的掌握程度。

期望得分 100 分。

Coin

算法零：

输出 0 即可。

算法一：

直接 $O(4^n)$ 暴力枚举，期望得分 20 分。

算法二：

贪心，问题所求的是最大化 $\min(\sum a_i - na - nb, \sum b_j - na - nb)$ ， na, nb 分别表示猜大猜小的个数。将 a_i, b_i 全部减一，所求变为 $\min(\sum a_i - nb, \sum b_j - na)$ 。我们枚举 na, nb ，贪心取最大的 na 个 a_i 和 nb 个 a_i 即可。

复杂度 $O(n^2)$ ，期望得分 60 分

算法三：

贪心+二分，假设 na 固定，那么 ai 的选取也是确定的。假设 nb 从 0 开始一个个往上增加， $\sum bj - na$ 递增， $\sum ai - nb$ 递减。 $\min(\sum ai - nb, \sum bj - na)$ 在 $\sum bj - na$ 超过 $\sum ai - nb$ 前递增。所以只需二分 $\sum bj - na$ 什么时候超过 $\sum ai - nb$ 即可。

复杂度 $O(n \log n)$ ，期望得分 100 分。

Travel

算法一：

如果没有那些土豪，这道题目显然是 SPFA。期望得分 30 分。

算法二：

如果有了这些限制，我们可以跑分层图 SPFA。即定义 $dis[T][u]$ 为在时间 T 的时候到达机场 u 的最早时间。转移到 $dis[T+1][u]$ 或者 $dis[T + ci][v]$ （对于每条 (u,v,ci) 的边）。

复杂度 $O(Ki * m)$ ，期望得分 80 分。

算法三：

运用贪心。显然，到达了某个机场之后，一定是在土豪一结束包场之后就飞走。

预处理一个 $Go[][]$ 数组， $Go[T][u]$ 表示在 T 的时间到达机场 u ，最近什么时间才能飞走。这个预处理的时间是 $O(n * Ki)$ 的。

然后跑 SPFA，将 SPFA 的转移方程改成： $dis[v] \leftarrow$

$Go[v][dis[u] + ci]$ 即可。

总复杂度 $O(m + n * K)$ ，期望得分 100 分。

算法四：

如果将时间点进行离散，即使时间点 $t_i \leq 10^9$ 也能做。

期望得分 100 +。