Day4 模拟赛解题报告

张志心 PaperCloud
浙江大学

2025 年 2 月 9 日

1. T1. 电子钟
2. T2. 成绩单
3. T3. 能量法阵
4. T4. 每日长跑

T1. 电子钟

简要题意

有一个环，你初始位于 t_0，指定了 n 个点 $t_1, \ldots, t_n$。每一步可以向任意方向走一个单位，问最少多少步可以走完所有点。

$1 \le n \le 5 \times 10^4, 0 \le t_i < 86400$

首先是对于时间的处理（包括读入和输出）

推荐的读入方法：

- 用 cin：`cin>>h>>ch1>>m>>ch2>>s;`
- 用 scanf：`scanf("%d:%d:%d",&h,&m,&s);`

推荐的输出方法：

- 用 cout：`setw(2),setfill('0'),cout<<h<<";"<<m<<":"<<s<<endl;`
- 用 printf：`printf("%02d:%02d:%02d",t/3600,t/60%60,t%60);`

T1. 电子钟

算法一（错误）

按一个方向走，直到走完所有的点。

反例：样例 2。

可以得到 60 分。

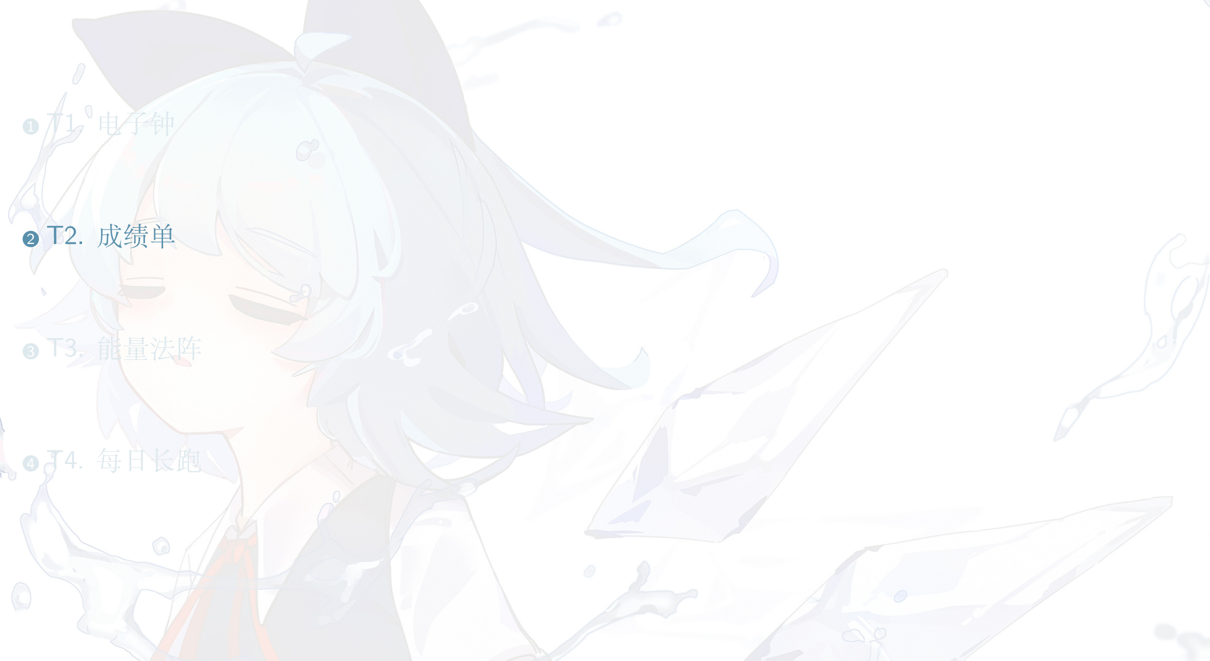
T1. 电子钟

算法二

最优的走法或者是按一个方向走完所有的点，或者是先按一个方向走到某个点，然后往回走直到所有点都走完。

所以我们枚举这个点即可。

时间复杂度 $O(n+T)$（桶排序）或 $O(n\log n)$（sort）。



有 n 个数，取出一个长度至少为 k 的连续子段，使它的中位数最大。

$1 \le k \le n \le 2 \times 10^5, 1 \le a_i \le 10^9$

T2. 成绩单

算法一

暴力枚举所有区间并求中位数。

时间复杂度 $O(n^3)$，期望得分 30 分。

T2. 成绩单

算法二

对 $k=1$ 的数据，直接输出全局最大值即可；

对 $k=n$ 的数据，直接输出全局中位数即可。

各 10 分，结合算法一可得 50 分。

算法三

二分答案 x。

将 $\geq x$ 的数全部视为 1，$< x$ 的数全部视为 -1。

则问题转化为是否存在长度大于等于 k 且和 > 0 的区间。

求前缀和后暴力枚举的时间复杂度为 $O(n^2 \log n)$，期望得分 60 分。结合算法二可得 80 分。

考虑优化算法三。

枚举区间的左端点 l，则我们只需判断是否存在一个右端点 $r \geq l + k$ 使得 $s_r > s_l$。这可以预处理后缀 max 后 $O(1)$ 查询。

时间复杂度降为 $O(n \log n)$。

1. T1. 电子钟
2. T2. 成绩单
3. T3. 能量法阵
4. T4. 每日长跑

n 个数排成一个环，每次可将相邻两个数合并得到它们的和，或者将相邻三个数合并得到 $x \times z - y$。求合并到一个数的最大值。

$1 \le n \le 35, |a_i| \le 10$

T3. 能量法阵

算法一

暴力搜索所有可能的合成方案（$n \geq 5$ 可以直接手玩）。

得分 ≤ 50。

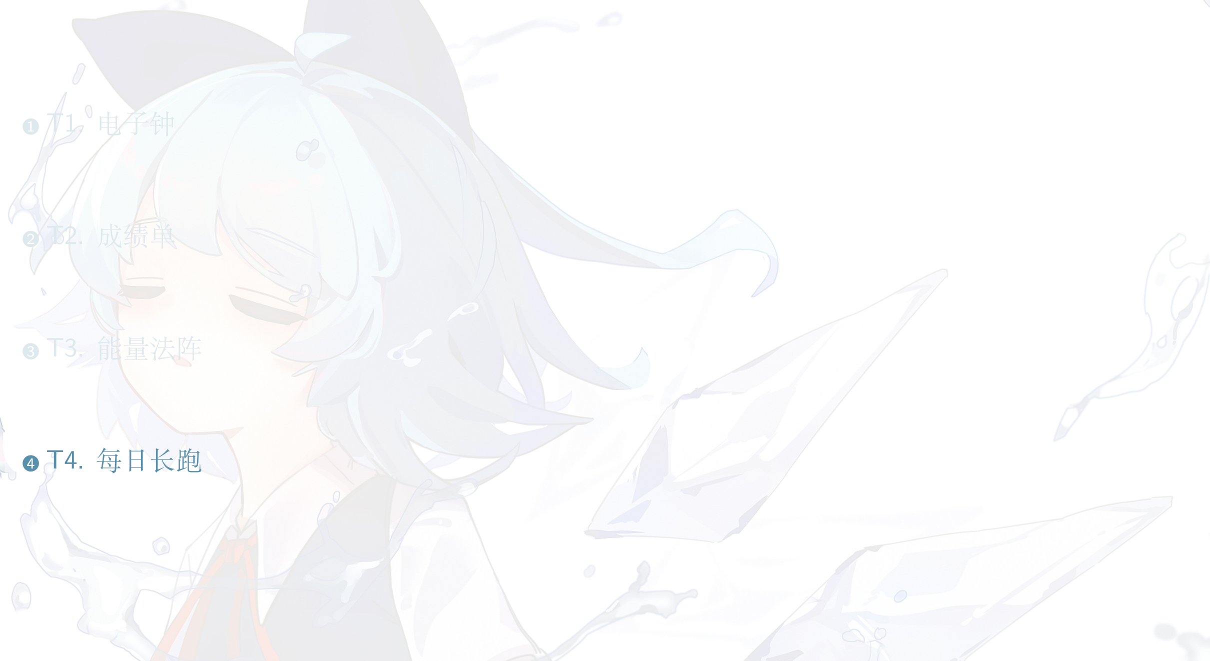
考虑区间 dp。类似【多边形游戏】一题，由于本题的合并有减法且数字有正有负，所以我们需要同时维护区间合并得到的最大值和最小值。故有转移

- $f(l,r) \leftarrow f(l,i) + f(i+1,r)$
- $g(l,r) \leftarrow g(l,i) + g(i+1,r)$
- $f(l,r) \leftarrow f(l,i)f(j+1,r) - g(i+1,j)$
- $f(l,r) \leftarrow g(l,i)g(j+1,r) - g(i+1,j)$
- $g(l,r) \leftarrow f(l,i)f(j+1,r) - f(i+1,j)$
- $g(l,r) \leftarrow f(l,i)g(j+1,r) - f(i+1,j)$
- $g(l,r) \leftarrow g(l,i)f(j+1,r) - f(i+1,j)$
- $g(l,r) \leftarrow g(l,i)g(j+1,r) - f(i+1,j)$

时间复杂度 $O(Tn^4)$。

考虑最极端情况 `10 -10 10 -10 ...`

此时最优策略为“二叉树式合并”，即每轮两两合并，合并四轮。最终答案约为 $(10^2+10)\times(((10^2+10)^2+10)^2+10)^2-10\approx 2.3\times 10^{18}<2^{63}$，因此可以在 `long long` 范围内存下。



T4. 每日长跑

简要题意

有一个背包和 n 种物品，每种物品有价值 v_i 和重量 w_i，第 i 种物品有 c_i 个，你要依次装这 n 种物品且保证任意时刻总重量在 $[-m, m]$ 内。求背包能装下的物品的最大价值和。

$1 \le n \le 10^3, 1 \le m \le 10^4, |w_i| \le 100, |v_i| \le 10^4, 1 \le c_i \le 10^3$

对于 $c_i = 1$ 且 a_i 全为正数的数据，可以发现这就是一个经典的 01 背包问题。

时间复杂度 $O(nm)$，空间复杂度 $O(m)$，期望得分 25。

T4. 每日长跑

算法二

对于 w_i 全为正数的数据，可以发现这就是一个经典的多重背包问题。暴力或二进制拆分有不同的部分分。

暴力时间复杂度 $O(nmk)$，空间复杂度 $O(m)$，期望得分 40。

二进制拆分时间复杂度 $O(nm\log k)$，空间复杂度 $O(m)$，期望得分 55。

T4. 每日长跑

算法三

二进制拆分后即转化为带负重量的 01 背包问题。

时间复杂度 $O(nm\log k)$，空间复杂度 $O(m)$，期望得分 100。

如果对转移顺序不清楚，可以直接多加一维表示当前天数，空间复杂度 $O(nm)$。