Book 解题报告

成都七中 王迪

Contents

1 题目大意

构造一个长度为 N 的整数数列 $\{S_N\}$，满足：

- $S_1 = X$；
- 对于 $1 < i \le N$，$S_i - S_{i-1}$ 要么为 $+A$，要么为 $-B$，其中 A, B 均为正整数；
- $\sum_{1 \le i \le N} S_i = M$。

对于 40% 的数据，$N \le 100$，$A, B \le 5$；
对于 60% 的数据，$N \le 1000$；
对于 100% 的数据，$N \le 10^5$。
数据保证有解。

2 算法分析

由于题目只要求构造一个可行解，所以很容易直接想到各种各样的贪心算法，但我所知的一些贪心算法都是不正确的，而盲目的贪心也不能帮助我们认清题目的本质。

所以我们考虑用数学语言来刻画这个问题：

令 $D_i = S_{i+1} - S_i$，其中 $1 \le i < N$，那么要么 $D_i = A$，要么 $D_i = -B$。

则 $S_i = X + \sum_{1 \le j < i} D_j$。我们尝试用 D_i 表示 M：

$$M = \sum_{1 \le i \le N} (X + \sum_{1 \le j < i} D_j)$$

令 $T = M - N \cdot X$，我们考虑上式每一个 D_i 出现的次数，则有：

$$T = \sum_{1 \le i < N} D_i \times (N - i)$$

此时，我们发现问题是把 1 到 $N - 1$ 的整数分成两个集合，一个集合中数之和乘以 A，另一个集合中数之和乘以 B，而这两个值的差为 T。

我们已经可以给出一些可以获得部分分的动态规划算法。

因为对 1 到 $N - 1$ 的每个整数有两个选择，我们考虑进行“降维”操作，即通过一些假设增加我们的已知量。

对所有 $1 \le i < N$ 令 $D_i = +A$，计 $Q = A \cdot \sum_{1 \le i < N} (N - i) = A \cdot \frac{N(N-1)}{2}$。

考虑 Q 和 T 的差值，而对于 D_i 若改变它为 $-B$ 则这个差值将减少 $(A + B)(N - i)$。

因为数据保证有解，故 $Q - T$ 必为 $(A + B)$ 的倍数，且记 $P = \frac{Q-T}{A+B}$，有 $0 \le P \le \frac{N(N-1)}{2}$。

于是问题变成了一个经典问题：从 1 到 $N - 1$ 中选择一些数，使得和为 P。

现在就可以比较轻松地设计算法了。

3 40% 的算法

因为 $N \le 100$，而 $P = O(N^2)$，所以我们可以设计一个动态规划算法。

记 $dp[i][j]$ 表示用 1 到 i 的数能否得到选一些和为 j，则有 $dp[i][j] = dp[i-1][j]\ or\ dp[i-1][j-i]$。

时间复杂度 $O(N^3)$，期望得分 40 分。

4　60% 的算法

直接的动态规划显然没有利用题目的特殊性：可用的数字是从 1 到 $N-1$，具有连续性！

容易设计出一个贪心的算法：每次从未选数中选择最大的一个不超过 P 的数，添加进方案中，然后迭代进行下去。

这个结论其实是显然的：

- 若 $P \leq N-1$，则可以直接构造出；
- 若 $P \geq N$，则我们的过程可以这样表示：$P = (N-1)+(N-2)+\cdots+(N-k)+R$。
  - 若 $R < N-k$，则可以直接构造；
  - 否则，可以把 $(N-K-1)$ 添加进方案，然后迭代下去。因为 $P \leq \frac{N(N-1)}{2}$，所以一定是可以构造出来的。

所以简单的两重循环，每次找到可用的最大数即可，时间复杂度 $O(N^2)$，期望得分 60 分。

5　100% 的算法

由上面的分析可以发现，我们构造出的方案一定是从 $N-1$ 开始递减连续的一段，再加上一个单独的值，所以完全可以 $O(N)$ 实现。期望得分 100 分。