1 Classical Problems

1.1 最长上升子序列

二分优化掉内层循环。注意到 LIS 的一个性质，如果当前的数比之前的某个数大，那么它的 LIS 长度比之前的那个数大。注意到之前 $O(n^2)$ 的算法的内层循环的问题。

明显地我们可以使用二分技术优化内层循环，维护一个数组 g[]，使得它单调，然后二分查找它的 LIS 的长度。

1.2 滑雪

dp[i][j] 表示到 (i,j) 这个格子的最长滑雪路线长度。

用推的方法：

$$dp[x][y] \to dp[i][j] |x - x'| + |y - y'| = 1, a[x'][y'] < a[x][y]$$

按照所有点的高度进行排序，然后按照高度由高到低枚举即可。

1.3 最长不互斥子序列

给定一个序列，找出最长不互斥子序列，即 $b[i] \,\mathrm{and}\, b[i-1] \neq 0$.

类似 LIS 的做法，维护一个 g[] 数组，使它表示满足不互斥性质的子序列的长度。

1.4 回文串划分

给一个字符串，划分成最少个回文子串。长度不超过 1000.

令 $dp[i]$ 表示已经将字符串的第 1 到 i 位处理完毕的最少划分次数。

$$dp[i] = \min\{dp[j] + 1\}, j < i, s[j, i]\text{是回文串}.$$

使用字符串 Hash 算法，判断是否是回文串（只需要对一个字符串正过来做一半的 Hash，倒着再做一遍 Hash，判断 Hash 是否相等。）

- 若回文串长度为奇数，可以预处理一个数组，表示以 i 点为中心的最长回文串长度 => $O(n^2)$.
- 若回文串长度为偶数，额外处理一个数组，表示以 i 和 i+1 为中心的最长回文串长度（使得 a[i]=a[i+1],a[i-1]=a[i-2]）=>$O(n^2)$.

1.5 传球问题

有 N 个人排成一个环，每个人选择向左或向右传球，最后一个拿到球的人输。问游戏进行 M 轮，第 i 个人输的方案数是多少。N 不超过 30，M 不超过 30。

dp[i,x] 表示经过了 i 轮，第 x 个人拿到球的方案数。

$$dp[i,x] = \min\{dp[i-1,p] + C(p,r)\}$$
$$dp[i][x] = \sum_{(p,r)} dp[i-1,p]$$

1.5.1 阶梯序列

B 序列是梯子序列，当且仅当：存在 x 使得

$$B(1) \leq B(2) \leq \cdots \leq B(x) \geq B(x+1) \geq \cdots \geq B(N).$$

给定一个序列 A，有 Q 次询问 $A(L \dots R)$ 是不是梯子序列.

$N, Q \leq 10^5$.

做一遍最长不降子序列+最长上升子序列。

预处理 up[i] 表示以 i 为起点向左最大有多少个单调上升的数，同样预处理 dowm[i] 数组表示向右有多少个单调上升的数，然后判断一下区间 [L,R] 内的 up 与 down 的长度。

1.6 区间染色

给定一个长度为 N 的序列，每一位有一个目标颜色，每次可以选择一个区间，将区间内的所有元素改为其目标颜色。设区间内不同颜色的数量为 X，则操作的代价为 X^2。求最小代价。$N \leq 5 \times 10^4$. 注意该点需要什么颜色就必须染成什么颜色，不能染成别的颜色。

很容易写出状态转移方程：

$$f[i] = \min\{f[j] + cost(j,i)\}, j < i;$$

1D1D 动态规划标准 DP 模型。

注意到直接输出 n 可以暴力骗分。