10月5日冲刺Noip2019模拟赛7(B班)
解题报告

题一：彩虹

对于100%的数据：$n,c \le 1000$。

- 把人看成点，在一个人和他喜欢的人之间连一条双向边。
- 如果一个人要乘上彩虹，那么和他连通的所有人都要乘上彩虹。
- 因此用并查集将一个人和他喜欢的人合并起来。
- 然后再跑一遍01背包即可。
- 时间复杂度 $O(nc)$。

题二：红十字

对于50%的数据： $1 \le n \le 500$。

- 合法的红十字要满足以下条件：
- 假设这个红十字的左上角是(i, j)，边长是2x + 1。
- 那么以(i, j + x)为左上角，(i + 2x, j + x)为右下角的子矩阵里面全部都是1。
- 也就是说这个子矩阵的权值和为 2x + 1。
- 同理以(i + x, j)为左上角，(i + x, j + 2x)为右下角的子矩阵的权值和也是 2x + 1。
- 此外，还要满足：以(i, j)为左上角，(i + 2x, j + 2x)为右下角的子矩阵的权值和是 4x + 1。
- 我们预处理出 sum[u][v] 表示以 (1,1) 为左上角，(u, v) 为右下角的矩阵的权值之和。
- 那么以 (x1, y1) 为左上角， (x2, y2) 为右下角的子矩阵的权值之和就是：
- sum[x2][y2] – sum[x2][y1 – 1] – sum[x1 – 1][y2] + sum[x1 – 1][x2 – 1]。
- 枚举所有的子矩阵，找到最大的合法子矩阵。
- 时间复杂度 $O(n^3)$。

题二：红十字

对于100%的数据： $1 \le n \le 2000$。

- 发现如果以 (x, y) 为**中心**，边长为 k 的子矩阵是合法红十字，那么以 (x, y) 为**中心**，边长为 $k-2$ 的子矩阵也是合法红十字。
- 那么我们枚举红十字的中心 (x, y)，求出中心为 (x, y) 的最大合法红十字的边长。
- 显然可以二分求这个边长。
- 时间复杂度 $O(n^2 \log n)$。

题三：柱状图

对于80%的数据： $2 \le n \le 7$。

- 全排列会写吧。
- 时间复杂度 O(数据组数 * n!)。

题三：柱状图

对于100%的数据： $2 \le n \le 15$。

- 全排列的题可以考虑状压dp。
- 发现答案就是：相邻两个数的差的绝对值之和+第一个数+最后一个数+2n。
- 我们考虑求 相邻两个数的差的绝对值之和+第一个数+最后一个数 的最大值。
- 答案涉及**相邻两个数的关系**。
- 因此设f[x][s]表示：用集合s中的长方形拼出的最大周长（不考虑上面那个2n）。
- g[x][s]表示：拼出最大周长的方案数。
- 集合s可以用二进制数表示，要保证x属于集合s。
- 当集合s中只有x这一个元素时，f[x][s]=a[x], g[x][s] = 1。
- 否则，枚举集合s中排在倒数第2个的长方形y，排在最后1个的长方形x。
- 记集合s除去元素x后的集合为t。
- if (abs(a[y] - a[x]) + f[y][t] > f[x][s]) f[x][s] = abs(a[y] - a[x]) + f[y][t], g[x][s] = g[y][t]。
- else if (abs(a[y] – a[x]) + f[y][t] == f[x][s]) g[x][s] += g[y][t]。

题三：柱状图

对于100%的数据： 2≤n≤15。

- 考虑怎么算答案：
- 令s=2^n-1，即s包含了1~n的所有元素。
- 记ans为最大周长，cnt为方案数。
- 枚举整张柱状图中的最后一个长方形x：
- if (f[x][s] + a[s] > ans) ans = f[x][s] + a[s], cnt = g[x][s]。
- else if (f[x][s] + a[s] == ans) cnt += g[x][s]。
- 时间复杂度 $O(2^n n^2$ * 数据组数)。
- 虽然 n = 15 的时候时间复杂度 O(1亿多)，但是在我这个慢死的电脑上只跑了0.078s。
- 显然常数很小。

题四：最长公共子序列

对于60%的数据：$1 \le n \le 1000$。

- 序列长度最多 5000。
- 考虑 dp：f[i][j] 表示 A[1…i] 和 B[1…j] 的最长公共子序列。
- 如果 A[i] = B[j]，$f[i][j] = \max(f[i-1][j], f[i][j-1], f[i-1][j-1]+1)$。
- 否则 $f[i][j] = \max(f[i-1][j], f[i][j-1])$。
- 时间复杂度 $O(n^2)$。

题四：最长公共子序列

对于100%的数据： $1 \le n \le 20000$。

- 关键在于：保证 1~n 这 n 个数在 A, B 中分别出现5次。
- 我们枚举 i，同时维护 f[j] 表示 A[1…i] 和 B[1…j] 的最长公共子序列。
- 并且要求这个最长公共子序列在 B 中必须以 B[j] 结尾。
- 一开始对于每个 x，用二维数组存下数值 x 在 B 中的所有位置。
- 枚举 i 时，枚举数值 A[i] 在 B 中的所有位置 k。
- 这时候所有的 f[j] 都是 A[1…i – 1] 和 B[1…j] 的最长公共子序列。
- 我们令 g[k] = max(f[1] ~ f[k – 1]) + 1，则 g[k] 就是 A[1…i] 和 B[1…k] 的最长公共子序列。
- 对于 B[j] != A[i] 的 j，显然 g[j] = f[j]，也就是说 f[j] 不用改动。
- 那么我们现在要做的就的单点修改，求前缀最大值。
- 用树状数组维护 f[j] 的前缀最大值。
- 时间复杂度 O(n log n)。