商汤在线编程挑战赛* — 题目讲解

skywalkert

2018 年 04 月 20 日

比赛情况

题目	通过情况 (通过/尝试)
A. 双人取数	59.5122% (122/205)
B. 我觉得海星	41.8972% (212/506)
C. 抽球游戏	62.8205% (049/078)
D. 最小？最大!	00.0000% (000/007)
E. 反函数求解	61.4035% (035/057)
F. 归并排序	20.0000% (003/015)
G. 危险路径	77.1144% (155/201)

* 共 2303 人报名，601 人提交，259 人通过至少一题

A. 双人取数 `Accepted 122 / Tried 205 / Submitted 376`

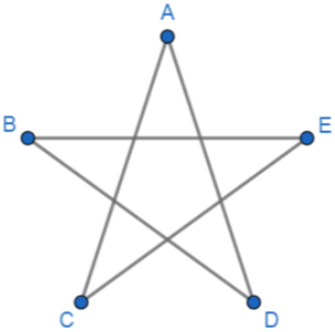
- 在 $n \times m$ 的矩阵中找一条从左上到右下的四连通路径、一条从右上到左下的四连通路径，最大化被至少一条路径覆盖的元素之和，元素之和可以是负数。
- 左上到右下的只能走右、走下，右上到左下只能走左、走下，路径必须包含矩阵的顶点。
- $3 \leq n, m \leq 1000$, 元素的绝对值不超过 1000。

A. 双人取数 `Accepted 122 / Tried 205 / Submitted 376`

- 路径公共部分只会算一次，预处理从矩阵四个顶点走到某个定点的最大元素之和，然后枚举公共路径。
- 公共路径只会是一条横线或一条竖线，按顺序枚举公共路径的一个端点，用之前的信息选出最优的另一个端点。
- 预处理和枚举公共路径都是 $\mathcal{O}(nm)$。

B. 我觉得海星 Accepted 212 / Tried 506 / Submitted 2123

- 给定 n 个点的简单无向图，判断这个图中是否存在一个环恰好经过 5 个互不相同的点。
- 给出的是图的邻接矩阵。
- 可能存在 $\mathcal{O}(n^2)$ 条边。
- $1 \leq n \leq 200$。



B. 我觉得海星 Accepted 212 / Tried 506 / Submitted 2123

- 做法 1：枚举 A, B, E，看 A, B 的公共邻居，A, E 的公共邻居，能否找出两个点做 C, D。

B. 我觉得海星 `Accepted 212 / Tried 506 / Submitted 2123`

- 做法 1：枚举 A, B, E，看 A, B 的公共邻居，A, E 的公共邻居，能否找出两个点做 C, D。
  - 使用 `bitset` 实现，$\mathcal{O}(n^4/w)$ $(w = 64)$。

B. 我觉得海星 Accepted 212 / Tried 506 / Submitted 2123

- 做法 1：枚举 A, B, E，看 A, B 的公共邻居，A, E 的公共邻居，能否找出两个点做 C, D。
  - 使用 bitset 实现，$\mathcal{O}(n^4/w)$ $(w = 64)$。
  - 只统计公共邻居个数、保留至多两个不同公共邻居，$\mathcal{O}(n^3)$。

B. 我觉得海星 `Accepted 212 / Tried 506 / Submitted 2123`

- 做法 1：枚举 A, B, E，看 A, B 的公共邻居，A, E 的公共邻居，能否找出两个点做 C, D。
  - 使用 `bitset` 实现，$\mathcal{O}(n^4/w)$ $(w = 64)$。
  - 只统计公共邻居个数、保留至多两个不同公共邻居，$\mathcal{O}(n^3)$。
- 做法 2：统计所有 5 个点的环数量，减去 2 个点的环与 3 个点的环共用点的子图数量，$\mathcal{O}(n^3)$。

B. 我觉得海星 `Accepted 212 / Tried 506 / Submitted 2123`

- 做法 1：枚举 A, B, E，看 A, B 的公共邻居，A, E 的公共邻居，能否找出两个点做 C, D。
  - 使用 `bitset` 实现，$\mathcal{O}(n^4/w)$ $(w = 64)$。
  - 只统计公共邻居个数、保留至多两个不同公共邻居，$\mathcal{O}(n^3)$。
- 做法 2：统计所有 5 个点的环数量，减去 2 个点的环与 3 个点的环共用点的子图数量，$\mathcal{O}(n^3)$。
- 做法 3：随机打乱边的顺序，`dfs` 得到一棵生成树，检查是否有非树边恰好覆盖 4 条树边，$\mathcal{O}(kn^2)$。

C. 抽球游戏 `Accepted 49 / Tried 78 / Submitted 192`

- 有 n 个有标号的球，每次从中有放回地抽取一个球，连续三次，定义这三次抽出的球的标号按位异或和为 x 的情况有 p_x 种。
- 给定 p_x，请你构造这 n 个球的标号，满足给定的限制，可能无解。
- $1 \leq n \leq 50$, 所有球的标号在 0 到 63 之间。

C. 抽球游戏 Accepted 49 / Tried 78 / Submitted 192

- 设标号为 y 的球有 c_y 个，则

$$p_x = \sum_{0 \le y_1, y_2, y_3 \le 63} [y_1 \oplus y_2 \oplus y_3 = x] c_{y_1} c_{y_2} c_{y_3}$$

- 沃尔什-阿达玛变换 $\mathrm{WHT}(c)$ 满足：$\mathrm{WHT}(a)$ 与 $\mathrm{WHT}(b)$ 等于 $\mathrm{WHT}(\mathrm{c})$ 当且仅当 $c_i = \sum_{j \text{ opt } k = i} (a_j \cdot b_k)$，其中 opt 可以为任一集合运算符。
- $\oplus$ 就是一种集合运算符。

C. 抽球游戏 Accepted 49 / Tried 78 / Submitted 192

- 结合 WHT 的相关性质可知，$\mathrm{WHT}(p) = \mathrm{WHT}^3(c)$，看上去只要暴力实现 WHT、逆 WHT 和开三次方根就做完了。

C. 抽球游戏 `Accepted 49 / Tried 78 / Submitted 192`

- 结合 WHT 的相关性质可知，$\mathrm{WHT}(p) = \mathrm{WHT}^3(c)$，看上去只要暴力实现 WHT、逆 WHT 和开三次方根就做完了。
- 需要判断无解：

C. 抽球游戏 `Accepted 49 / Tried 78 / Submitted 192`

- 结合 WHT 的相关性质可知，$\mathrm{WHT}(p) = \mathrm{WHT}^3(c)$，看上去只要暴力实现 WHT、逆 WHT 和开三次方根就做完了。
- 需要判断无解：
  1. 开三次方根后是否为整数？

C. 抽球游戏 `Accepted 49 / Tried 78 / Submitted 192`

- 结合 WHT 的相关性质可知，$\mathrm{WHT}(p) = \mathrm{WHT}^3(c)$，看上去只要暴力实现 WHT、逆 WHT 和开三次方根就做完了。
- 需要判断无解：
  1. 开三次方根后是否为整数？
  2. 逆 WHT 后是否为整数？

C. 抽球游戏 `Accepted 49 / Tried 78 / Submitted 192`

- 结合 WHT 的相关性质可知，$\mathrm{WHT}(p) = \mathrm{WHT}^3(c)$，看上去只要暴力实现 WHT、逆 WHT 和开三次方根就做完了。
- 需要判断无解：
  1. 开三次方根后是否为整数？
  2. 逆 WHT 后是否为整数？
  3. $\sum_{0 \le y \le 63} c(y)$ 是否等于 n？

C. 抽球游戏 `Accepted 49 / Tried 78 / Submitted 192`

- 结合 WHT 的相关性质可知，$\mathrm{WHT}(p) = \mathrm{WHT}^3(c)$，看上去只要暴力实现 WHT、逆 WHT 和开三次方根就做完了。
- 需要判断无解：
  1. 开三次方根后是否为整数？
  2. 逆 WHT 后是否为整数？
  3. $\sum_{0 \le y \le 63} c(y)$ 是否等于 n？
  4. ~~行末是不是多输出了空格？~~

D. 最小？最大！ Accepted 0 / Tried 7 / Submitted 16

- 构造字典序最大的 S 使得 S 的字典序不大于它的所有循环表示，且每种字符在 S 中的数量为给定值。输出 S 的 hashCode 和其中 m 个位置的字符。
- 循环表示指将字符串开头的一部分字符依次移动到字符串末尾得到的新字符串。
- $1 \leq m \leq 10^5$, S 的长度不超过 10^{18}。

D. 最小？最大！ Accepted 0 / Tried 7 / Submitted 16

先看几个小例子：

- 重排 abc 的答案是 acb
- 重排 aabbb 的答案是 ababb
- 重排 aaabb 的答案是 aabab
- 重排 aaabc 的答案是 aacab
- 重排 acecoder 的答案是 aroeedcc

D. 最小？最大！ Accepted 0 / Tried 7 / Submitted 16

- 维护可重复字符串集合 P，初始只有长度为 1 的字符串，每个字符出现给定次数。
- 每次从中删去字典序最小的字符串 A、字典序最大的字符串 B，把 A 与 B 的拼接加入 P，最终剩下的一个字符串就是所求的 S。
- 这里略去证明，默认你知道这题~~是拟阵~~可以贪心了。

D. 最小？最大！ Accepted 0 / Tried 7 / Submitted 16

- 不妨设字典序最小的字符串 A 在 P 中有 x 个，字典序最大的字符串 B 在 P 中有 y 个，则接下来 $\min(x, y)$ 次操作的效果是相同的。

D. 最小？最大！ Accepted 0 / Tried 7 / Submitted 16

- 不妨设字典序最小的字符串 A 在 P 中有 x 个，字典序最大的字符串 B 在 P 中有 y 个，则接下来 $\min(x, y)$ 次操作的效果是相同的。
- 这样 $\min(x, y)$ 次操作相当于将某种串变成另一种串，将另一种串的个数变成 $\max(x, y) - \min(x, y)$ 个，类似辗转相减的过程，注意串的字典序可能变化。

D. 最小？最大！ Accepted 0 / Tried 7 / Submitted 16

- 不妨设字典序最小的字符串 A 在 P 中有 x 个，字典序最大的字符串 B 在 P 中有 y 个，则接下来 $\min(x, y)$ 次操作的效果是相同的。
- 这样 $\min(x, y)$ 次操作相当于将某种串变成另一种串，将另一种串的个数变成 $\max(x, y) - \min(x, y)$ 个，类似辗转相减的过程，注意串的字典序可能变化。
- 操作可以用辗转相除优化，S 可以用 $\mathcal{O}(26 \log_{26} |S|)$ 个点的有向无环图进行压缩表示。可能注意下等比数列求和公式就可以随便算算了。

E. 反函数求解 `Accepted 35 / Tried 57 / Submitted 206`

- 令 $\text{digitSum}(x)$ 表示 x 的十进制各位数字之和。
- 构建一个图，有无穷个点，编号 $0, 1, \cdots$，使每个点 x 向 $x + \text{digitSum}(x)$ 连一条有向边。
- 求出所有 x 使得从点 x 出发恰好走 m 步能到达点 n。
- $0 \le n \le 10^{18}$, $0 \le m \le 10^5$。

E. 反函数求解 `Accepted 35 / Tried 57 / Submitted 206`

- 当 $0 \le x \le 10^{18}$ 时，$\text{digitSum}(x) \le 18 \times 9 = 162$。

E. 反函数求解 `Accepted 35 / Tried 57 / Submitted 206`

- 当 $0 \le x \le 10^{18}$ 时，$\text{digitSum}(x) \le 18 \times 9 = 162$。
- 直接扫一遍 $n - 162m \le x \le n$ 的所有点，算出 $\text{digitSum}(x)$ 递推一下就行了。

E. 反函数求解 `Accepted 35 / Tried 57 / Submitted 206`

- 当 $0 \le x \le 10^{18}$ 时，$\text{digitSum}(x) \le 18 \times 9 = 162$。
- 直接扫一遍 $n - 162m \le x \le n$ 的所有点，算出 $\text{digitSum}(x)$ 递推一下就行了。
- 需要均摊线性计算 $\text{digitSum}(x)$。

E. 反函数求解 `Accepted 35 / Tried 57 / Submitted 206`

- 当 $0 \le x \le 10^{18}$ 时，$\mathrm{digitSum}(x) \le 18 \times 9 = 162$。
- 直接扫一遍 $n - 162m \le x \le n$ 的所有点，算出 $\mathrm{digitSum}(x)$ 递推一下就行了。
- 需要均摊线性计算 $\mathrm{digitSum}(x)$。
  - 做法 1：模拟 x 变成 $x+1$ 的过程，顺便维护 $\mathrm{digitSum}(x)$，均摊线性。

E. 反函数求解 Accepted 35 / Tried 57 / Submitted 206

- 当 $0 \le x \le 10^{18}$ 时，$\mathrm{digitSum}(x) \le 18 \times 9 = 162$。
- 直接扫一遍 $n - 162m \le x \le n$ 的所有点，算出 $\mathrm{digitSum}(x)$ 递推一下就行了。
- 需要均摊线性计算 $\mathrm{digitSum}(x)$。
  - 做法 1：模拟 x 变成 $x + 1$ 的过程，顺便维护 $\mathrm{digitSum}(x)$，均摊线性。
  - 做法 2：把 x 分成三段长度均匀的部分，预处理 $\mathrm{digitSum}(x)$ $(0 \le x \le 10^6)$。

E. 反函数求解 `Accepted 35 / Tried 57 / Submitted 206`

小贴士：

- 可能需要注意下输出是对 2^{64} 取模。
- 可能需要注意下 $n = 0$ 和 $m = 0$ 的情况。

F. 归并排序 Accepted 3 / Tried 15 / Submitted 23

- 给出一个传统归并排序算法，在合并两个序列时每次弹出的是当前两个序列的头部的最小值。
- 在此基础上增加限制：递归层数达到 k 时，不再划分序列，直接返回未排序的序列。
- 统计有多少 1 到 n 的排列在经过新的排序算法后，能删掉一个元素，使剩下元素递增。数量对 p 取模。
- $1 \leq n, k \leq 50$, $10^8 \leq p \leq 10^9$, p 是质数。

F. 归并排序 `Accepted 3 / Tried 15 / Submitted 23`

- 抽取第 k 层所有可能乱序的区间，分析归并算法可知，满足条件的排列至多在一个区间内有一个畸形点。

F. 归并排序 Accepted 3 / Tried 15 / Submitted 23

- 抽取第 k 层所有可能乱序的区间，分析归并算法可知，满足条件的排列至多在一个区间内有一个畸形点。
- 若畸形点比前一个点小，则之后与有序序列归并还会得到相同性质的序列。

F. 归并排序 `Accepted 3 / Tried 15 / Submitted 23`

- 抽取第 k 层所有可能乱序的区间，分析归并算法可知，满足条件的排列至多在一个区间内有一个畸形点。
- 若畸形点比前一个点小，则之后与有序序列归并还会得到相同性质的序列。
- 若畸形点比前一个点大，则之后小于它的点必须有序地直接排在它的后面。

F. 归并排序 Accepted 3 / Tried 15 / Submitted 23

- 抽取第 k 层所有可能乱序的区间，分析归并算法可知，满足条件的排列至多在一个区间内有一个畸形点。
- 若畸形点比前一个点小，则之后与有序序列归并还会得到相同性质的序列。
- 若畸形点比前一个点大，则之后小于它的点必须有序地直接排在它的后面。
- 对于长度为 L 的区间，第一部分可以 $\mathcal{O}(1)$ 算，第二部分可以 $\mathcal{O}(L)$ 算，总复杂度线性。

~~算漏第二种情况能过样例，想不到吧!~~

G. 危险路径 Accepted 155 / Tried 201 / Submitted 574

- 给定 n 个点、m 条边的连通无向图，点编号 $1, 2, \cdots, n$，边有权值。
- 对于每个点 u，找到一条从 u 到 v 的路径，最小化路径上边的权值最大值，记为 $d(u, v)$。
- 令 $f(u) = \sum\limits_{u \neq v} d(u, v)$，求 $\bigoplus\limits_{u=1}^{n} (u \cdot f(u))$。
- $1 \leq n \leq 10^5$, $0 \leq m \leq 3 \times 10^5$, 边权非负且不超过 10^9。

G. 危险路径 Accepted 155 / Tried 201 / Submitted 574

- $d(u, v)$ 对应的路径一定在整个图的最小生成树上，不同的最小生成树可以得到相同的答案。

G. 危险路径 `Accepted 155 / Tried 201 / Submitted 574`

- $d(u, v)$ 对应的路径一定在整个图的最小生成树上，不同的最小生成树可以得到相同的答案。
- 考虑 `Kruskal` 算法过程，在添加一条权值为 w 的边连通点集 S 和 T 时，S 中每个点的 f 会增加 $|T| \cdot w$，而 T 中每个点的 f 会增加 $|S| \cdot w$。

G. 危险路径 `Accepted 155 / Tried 201 / Submitted 574`

- $d(u,v)$ 对应的路径一定在整个图的最小生成树上，不同的最小生成树可以得到相同的答案。
- 考虑 `Kruskal` 算法过程，在添加一条权值为 w 的边连通点集 S 和 T 时，S 中每个点的 f 会增加 $|T| \cdot w$，而 T 中每个点的 f 会增加 $|S| \cdot w$。
- 只考察了并查集的基本应用，可以做到 $\mathcal{O}(n\alpha(n) + m \log m)$，启发式合并可能会超时。

G. 危险路径 `Accepted 155 / Tried 201 / Submitted 574`

- $d(u, v)$ 对应的路径一定在整个图的最小生成树上，不同的最小生成树可以得到相同的答案。
- 考虑 `Kruskal` 算法过程，在添加一条权值为 w 的边连通点集 S 和 T 时，S 中每个点的 f 会增加 $|T| \cdot w$，而 T 中每个点的 f 会增加 $|S| \cdot w$。
- 只考察了并查集的基本应用，可以做到 $\mathcal{O}(n\alpha(n) + m \log m)$，启发式合并可能会超时。
- `Trick`：答案可能超过 $2^{63} - 1$，但是不会超过 $2^{64} - 1$。

讲解结束

Q & A