Potyczki Algorytmiczne 2017

Syloviaely
Peking University

Zasady Gry

- 分为线上赛和线下赛（这儿只讲线上赛）
- 线上赛一共 12 题，分为AB两组各 6 道
- 类似 OI 赛制

Zasady Gry

#	User	per	ska	ilo	zap	moz	pra	dzi	sku	ban	car	gie	osa	Sum
1	jiry_2	10	10	10	10	10	10	10	10	10	10	10	10	120
2	Maciej Wawro	10	10	10	10	10	10	10	10	10	10	8	3	111
3	Marek Sommer	10	10	10	10	10	10	10	8	9	10	0	10	107
4	Mateusz Radecki	10	10	10	10	10	9	10	4	10	10	8	6	107
5	Krzysztof Maziarz	10	10	10	10	10	10	10	1	10	10	10	2	103
6	Kamil Dębowski	10	10	10	10	10	10	10	8	10	9	6		103
7	Krzysztof Potępa	10	10	10	10	10	10	10	0	10	6		10	96
8	Jan Tabaszewski	10	10	10	10	10	10	7	4	10	10		2	93
9	Jarosław Kwiecień	10	10	10	10	10	10	7	4	10	9			90
10	Radosław Serafin	10	10	10	10	10	10	10	0	10	6		1	87
11	Michał Górniak	10	10	10	10	10	10	10		10	6		0	86
12	Wojtek Nadara	1	10	10	10	10	0	10	2	10	10	8	2	83
13	michaltadeusz	10	10	10	10	10	10	10	0	10			2	82

Mały Prezent.

- 准备了一些 Steam 的小游戏

Mały Prezent.

Mały Prezent.

1B Skarbonka

- 给出 n 个数 ai
- 令 K 为 2^ai 之和
- 找到最大的 b 使得 K >= 2^b
- n <= 1e6 ai <= 201718

1B Skarbonka

- 让大家体验一下
- 是真的有 Steam 游戏送的

2B Zapiekanki

- 你是一个小餐厅的老板，你从 0 时刻开始工作
- 有 n 个人来你这儿吃饭，第 i 个人 t_i 时刻到达
- 你烧一份饭需要 d 单位时间，烧饭不能中断且烧好后必须马上给人吃掉
- m 次询问，每次给出 d 问最少的总等待时间
- $n,m \le 2e5$ $t_i \le 1e12$ $d \le 1e6$

2B Zapiekanki

- 你的做菜过程肯定是连续的一段一段
- 如果 [l,r] 的人为连续的一段，那么菜的完成时间是 t[l]+id，人到达时间是 t[i]，求和相减即等待时间
- 从小到大枚举 d，维护连续段，每一次只可能是两个连续段合并成一段，用并查集可以轻易的维护
- O(maxd + n)

5B Banany

- 一棵树，有点权和边权，一条路径 (u,v) 的权值定义为 v 的点权减去路径的边权和
- 你第 0 天在 1 号点，每天你都会找一条从当前点出发的权值最大路径走过去
- 同时每天会发生一次变化，单点修改点权或者边权
- 模拟这个过程，即输出每天所在的点
- n,q <= 100000

5B Banany

- 直接点分树
- 需要搞个线段树来维护权值的修改
- $O(n \log^2 n)$

5B Carcassonne

- 一个 n*n 的黑白网格，黑色部分四联通
- 你需要把 K 个白色位置染黑，使得黑色仍联通
- 问方案数，取模
- n <= 3000 K <= 4

5B Carcassonne

- 如果不考虑最开始的黑色部分，填进去的黑色部分一定是一个一个的联通块
- 预处理所有大小 <= 4 的联通块的形状
- 每一个联通块至少有一个格子和原来的“主”联通块相邻
- 怎么不重不漏?

5B Carcassonne

- 喜闻乐见的大分类讨论
- 令与初始黑块相邻的格子为关键格
  - 四个格子都是关键格
  - 恰有一个非全关键格的联通块
  - 有两个大小为 2 的非全关键格的联通块

2A Lloczyn

- 给出一个质数集合 S = {Pi}
- 一个数是好的当且仅当它的所有质因子都在集合中
- 求小于等于 N 的最大的好的数
- N <= 1e18 Pi <= 100, 8 M 空间 6s

2A Lloczyn

- <= 100 只有 25 个质数
- meet in middle
- 均匀选取一半的质数，大概只能生成 1e7 个可能的数
- 排序后二分查找

2A Lloczyn

- 太慢?
- 绝大部分数都 >= 1e12，故绝大部分查询都询问 <= 1e6 最大的数
- 线性预处理所有这类询问 O(1) 查询

2A Lloczyn

- 空间太大?
- meet in middle 的时候一定有一边 <= 根号
- 所以限定第一部分 <= 根号，这样存下来的数就少了，然后交换两个集合再做一遍

1A Permutacja

- https://sio2.mimuw.edu.pl/contest/

1A Permutacja

- 一个长度为 n 的排列是好的当且仅当它的逆序对个数为 n*(n-1)/4
- 给出 n 和 K，问字典序第 K 小的长度为 n 的好的排列
- n <= 250000 K <= 1e18

1A Permutacja

- K 的 1e18 相对 n! 是一个非常小的数
- 令 dp[i][j] 表示长度为 i 逆序对个数恰好为 j 的排列个数，dp[i] 以 $j = i*(i-1)/2$ 对称
- 猜测只有当 i 和 j 都很小的时候，dp[i][j] 才能小于等于 1e18

1A Permutacja

- 可以暴力 dp 出所有小于等于 1e18 的 dp[i][j]
- 按位枚举，需要用一个树状数组来找当前第 k 大的值
- O(n log n)

3A Praca domowa

- 一个长度为 n 的数列，进行了 m 次单点修改
- 把第 i 次修改后的序列记为 Ai
- 将 Ai 按照字典序排序（输出排序后的下标）
- n,m <= 5e5 数字大小 1e9

3A Praca domowa

- m 个版本的数列可以用主席树得到
- 比较两个版本的大小是 $O(\log n)$ 的
- 直接快速排序 $O(n \log^2 n)$

3A Praca domowa

- 在主席树上归并，对同一个位置不同版本的节点排序
- 每一次用左儿子的相对顺序和右儿子的相对顺序作基数排序
- 时间复杂度 O(节点数) 即 $O(n \log n)$

3B Mozaika

- 有 n 个和坐标轴平行的正方形（边长可能不同），它们互不重叠且恰好组成了一个长方形
- 现在只给出正方形的左下角（长方形形态也未知），你需要给出一个边长的合法方案或者输出无解
- n<=2000 T<=50 坐标范围 1e9

3B Mozaika

- 矩形的左边界和下边界可以很容易确定
- 问题1：怎么确定矩形的右边界和上边界
- 问题2：怎么把正方形填进一个固定的矩形

3B Mozaika

- 矩形左上角的正方形是确定了的
- 枚举这个正方形右侧的正方形，则上边界就确定了
- 如果这个正方形右侧没有正方形，则把这个正方形删去，剩下的正方形的上边界就确定了

3B Mozaika

- 确定了上边界之后，可以从左上角的正方形开始，依次放入紧贴上边界的正方形
- 这样可以在 O(n) 时间内唯一确定一个右边界
- 可以在 O(n^2) 的时间内确定 O(n) 个可能的矩形边界

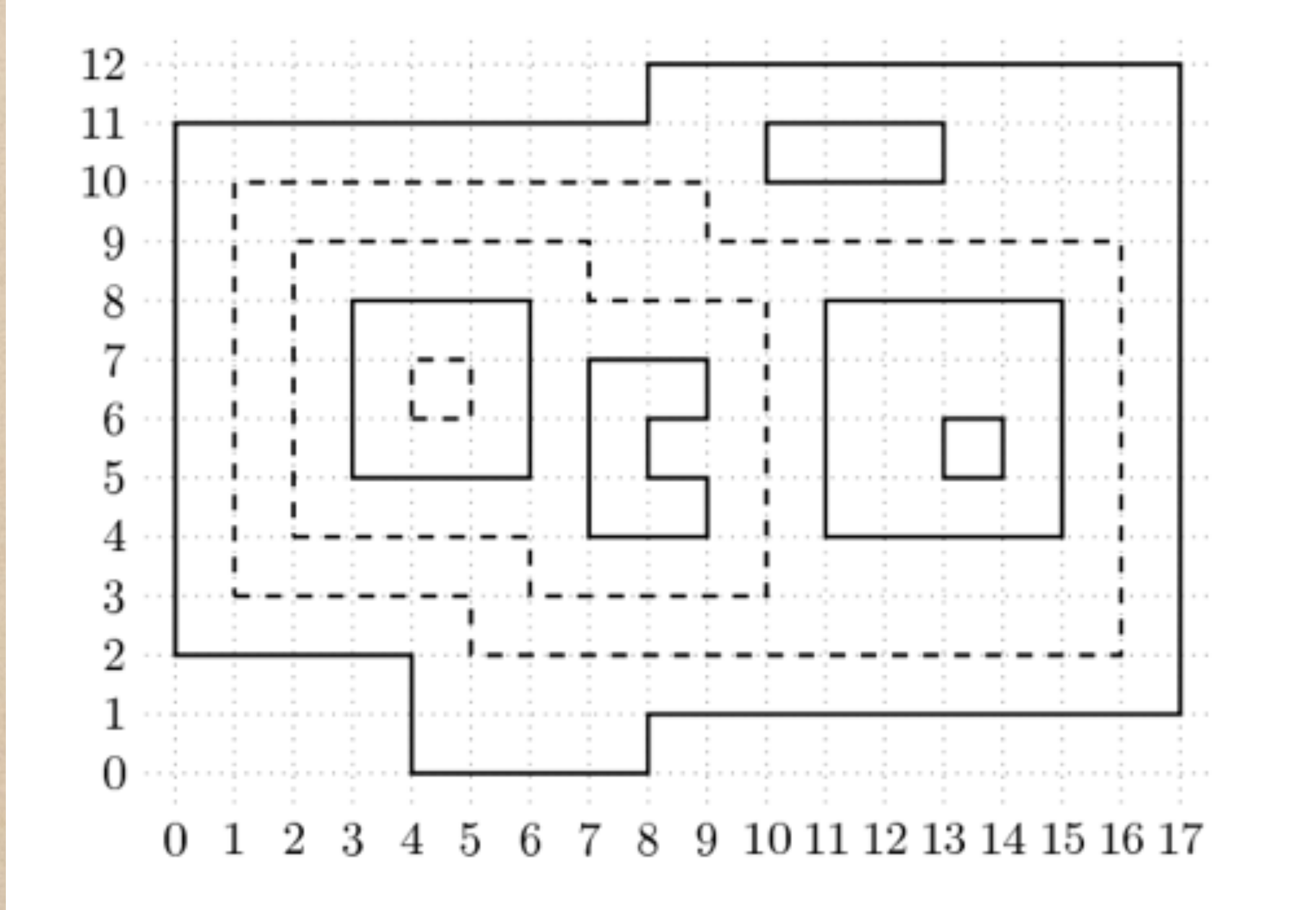
3B Mozaika

- 需要在 O(n) 复杂度内判断是否可行
- 从上到下填入正方形，维护上轮廓
- 每一次考虑高度相同的一条线段，它的左端点下方必须放入一个正方形，这个左端点已经唯一确定
- 重复这个过程直到所有正方形都被放入
- 用一个哈希表维护坐标可以直接模拟

5A Giewont

- 给出若干条整数坐标的闭合曲线，满足每条曲线都和坐标轴平行且曲线两两不交，保证存在一条包含所有其他曲线的曲线 s
- 你可以再加上若干条满足上述条件的闭合曲线
- 问在 s 内的曲线最多能嵌套几层（包括 s）
- 总点数 50000 坐标范围 1e8

5A Giewont



- 虚线为自己添加的线，最大答案为 5

5A Giewont

- 每一个格子都有一个权值 H，八联通的格子权值差距至多为 1
- 同一条曲线内部边界的格子权值相同，外部边界权值相同，内部比外部严格大 1
- 最外围曲线边界格子权值全为 0
- 问权值最大值能是多少

5A Giewont

- 问题 1：求每一条曲线外部边界格子的最大海拔
- 问题 2：求出所有格子的最大海拔

5A Giewont

- 考虑曲线之间的权值限制关系
- 考虑曲线 i 和曲线 j，i 的外部边界格子到曲线 j 的内部边界格子的最近距离为 d
- w[i] - d <= w[j] <= w[i] + d

5A Giewont

- 可以直接 O(n^2) 求出每一个曲线的 w[i]
- 如何通过 w[i] 计算得到答案?
- 考虑只有一个封闭曲线的情况。

5A Giewont

- 二分答案，相当于判断一堆矩形是否已经覆盖了一个封闭曲线
- 扫描线
- O(n^2+nlogn)
- wys一下还是能通过这个题的

5A Giewont

- 瓶颈在求 w[i]，想办法优化这一部分
- 求两个封闭曲线之间的距离，只需要考虑同方向之间的线段更新
- 相当于 m 个三维点，两两之间连边权为切比雪夫距离的无向边，求最短路
- 变成四维的 kd-tree O(n^7/4)

5A Osady i warownie

- 一个 n * m 的棋盘，有一些位置是障碍，有一些位置上有人，保证非障碍点四联通
- 在线进行若干次操作：
  - 将一个人移动到一个相邻位置
  - 把一个位置变成障碍，如果这个操作会使一对人无法互相到达了，就不进行这个操作。输出 01 表示是否成功
- n,m <= 1000 操作次数不超过 1e6

5A Osady i warownie

- 考虑棋盘的对偶图
  - 加点相当于在对偶图上加边
  - 原图的联通块相当于对偶图上的一个环
- 任意两人都能互相到达的条件等价于每个环内要么没人要么包含所有人

5A Osady i warownie

- 如果人不移动，那么所有人向右引一条射线
- 给所有人随机一个权值，射线权值为人的权值
- 每一条边的权值为经过这条边的射线权值的异或和，那么每一个联通块的人的权值的异或和相当于这个环所有边的异或和

5A Osady i warownie

- 动态加边维护环是并查集的经典问题
- 维护对偶图的一颗生成树
- 令所有人的权值异或和为 T
- 非树边的权值必须为对应路径的权值异或和或者权值异或和再异或上 T

5A Osady i warownie

- 假设人会移动
- 那么保持射线的初始部分不变，把开头部分移动成折线即可
- 时间复杂度 $O(q\alpha(n))$

4B Działka

- 这是一个分布式题
- 你的程序会被若干个线程同时运行，最终的运行时间由运行最慢的线程决定
- 所以要尽量提高程序的并行性
- 你可以用一些接口来在线程之间传递信息：

4B Działka

- ```
  int NumberOfNodes();
  int MyNodeId();
  ```

  这两个函数表示总共的 node 个数和当前计算机的编号。

- ```
  void PutChar(int target, char value);
  void PutInt(int target, int value);
  void PutLL(int target, long long value);
  ```

  这三个函数表示讲对应的内容传输到对应的计算机上。

- ```
  void Send(int target);
  ```

  当 put 的时候会把信息先放到 buffer 里面，只有当 send 的时候才会将信息真正的传出去，并且清空 buffer。
  ```

4B Działka

- ```
  int Receive(int source);
  ```
  ```
  char GetChar(int source);
  int GetInt(int source);
  long long GetLL(int source);
  ```
  对应的接收操作。
- 一般来说一次 send 和 receive 需要 $5 * 10^{-3}$s 的时间。
- 可以使用库在本地搭起简单的分布式环境。
  ```

4B Działka

- 通信的接口限制了一些行为:
  - 无法知道消息长度
  - 不把消息读完会 RE
  - 读的长度超过了消息的长度会 RE
  - 传递不定长的信息?

4B Działka

- 一个简单的小例题:
  - 有一个 1e9 长度的数列，你可以使用给出的接口访问某一个位置的值
  - 你需要求出所有数的和
  - 你可以使用 100 个线程
- 将序列分成 100 段，每一个线程求一段的和并发送给0号线程累加答案并输出

4B Działka

- 一个 n * m 的 01 网格，你可以用给出的接口访问某一个位置的值
- 问有多少个全 1 矩形
- n,m <= 75000, 100 个线程

4B Działka

- 如何划分问题?
- 每一个线程处理连续的若干列，计算右边界在这些列中的矩形个数
- 需要什么信息?
- 每一行在边界左边的连续 1 的个数 L[i]

4B Działka

- 串行传递?
- 并行传递!
- 已得 L[i] 怎么计算答案?
- 单调栈

4A Skup akcji

- 这也是个通信题
- 但是有一些线程对 (u,v)，他们的单向通信被阻断了，所有从 u 到 v 发的信息都会变成0。但是保证了存在一个线程可以直接或者间接接受到所有线程的信息

4A Skup akcji

- 通信的接口限制了一些行为:
  - 无法知道消息长度
  - 不把消息读完会 RE
  - 读的长度超过了消息的长度会 RE

4A Skup akcji

- 你控制了若干个线程，每个线程有一些信息，你需要把这些信息汇总到一个线程里
- 线程数 <= 100，每个点发送信息不超过 1200

4A Skup akcji

- 对每一个点 i，可以求出哪些点可以发消息过来
  - 所有点向其他点发一个 1
- 暴力做法?

4A Skup akcji

- 每一个点向其他点发一遍自己的信息
- 变成 0 的时候无法知道有多长，无法读完
- 改进?
- 已知了入边，利用入边“申请”信息。

4A Skup akcji

- 暴力做法:
  - 求出入边
  - 向入边发送一个 0 然后开始监听
  - 接受到来自 i 的 0 就把信息发给 i
  - 一旦集齐 n 颗龙珠就召唤神龙

4A Skup akcji

- 小细节：如何保证只输出一次?
  - 输出后给所有点发一个消息
    - 有可能拦不住早就输出了
  - 从 1 到 n 考虑每一个位置，如果不能输出就给下一个位置发一个消息来激活
    - 无法确定是否不能输出

4A Skup akcji

- 想办法找到一颗支撑树，那么就可以硬点根来输出，直接套用暴力做法，O(n) 次传递
- 简单情况：根为 1 号
- 倒着 BFS

4A Skup akcji

- 1处于态2，其他点属于态1
- 态1的点收到0后进入态2，且来源点为父亲
- 态2的点向所有入边发送一个0，进入态3
- 态3的点“忽略”所有的0
- 每个点O(n)次可以求出反向树

4A Skup akcji

- 如何统计答案?
- 每一个态 3 节点向父亲发送自己的信息（用 1 开头），如果接受到了信息则再转发给父亲
- 节点 1 集齐信息后输出并给所有节点发送 K 个 0 表示结束

4A Skup akcji

- 根节点不为 1 的情况?
- 怎么判断 1 不是根?
- 当一个点被访问到的时候，向所有点发送一个 0 表示它被访问了
- i 第一次接受到 j 发来的 0 时表示 j 被访问了，然后再按照之前的算法转移态

4A Skup akcji

- 如果 1 的所有出边都已经把信息返回了还没有集齐，那么就从下一个位置还没有被访问的位置作为根，发送一个 0 开始 DFS
- 如果还没有结束就找再下一个，直到有一个位置集齐所有信息后输出，并给全部节点发 K 个 0 结束

4A Skup akcji

- 1 开始 DFS （发送 0 DFS）
- i 第一次收到 j 的 0 时表示 j 被访问了，第二次收到表示 j 是父亲，并开始收集子树信息传给 j
- 如果 1 还没有结束就切换根
- 一旦有节点集齐就输出并退出
- 有一万个细节，每一个节点只有 O(n) 次操作

Potyczki Algorytmiczne 2017

- Thank you sir.