基础算法

分治、贪心、倍增和构造

2019 年 8 月 2 日

黄哲威 hzwer

自我介绍

- 北京大学16级计算机科学方向
- 计算概论 A，数据结构与算法 A ，算法设计与分析讨论班助教
- NOI银牌，CTSC金牌，ACM区域赛金牌
- 旷视科技(Megvii) Research Intern 研究计算机视觉与强化学习
- hzwer.com

本节目标

- 基本的二分算法，分治算法，贪心算法，倍增算法

回顾经典模型

通过刷题锻炼识别题目类型的能力

前置技能 - 简单数据结构

- 二叉堆，可以在 O(logn) 的时间内插入删除，O(1) 的时间查询最小值，要求掌握 priority_queue
- 平衡树，O(logn) 插入删除，O(logn) 查询一个元素的前驱后继，要求掌握 set 和 multiset，也可以用于查询最大最小值
- map 可以当一个哈希表，使用形如一个下标范围扩大的数组
- 参考 https://wenku.baidu.com/view/93f33b3b192e45361066f5eb.html

分治

- 分成相同或相似的子问题, 子问题可简单的直接求解，原问题的解即子问题的解的合并

- 注意复杂度计算，考虑分治每一层的开销，不要把问题想复杂

- 复杂度分析：

- 递归树：$T(n) = kT(n/m) + f(n)$

对于归并排序：$T(n) = 2T(n/2) + O(n) \Rightarrow O(n \log n)$

$T(n) = T(n/2) + O(n) \Rightarrow O(n)$

$T(n) = 2T(n/2) + O(n^2) \Rightarrow O(n^2)$

快速幂

- 快速幂 a^b
- 预处理 a^1 a^2 a^4 … a^(2n)，对 b 做二进制拆分
- 如 a^21 = a^16 * a^4 * a^1

例题.逆序对

- 设 A 为一个有 n 个数字的数列，其中所有数字各不相同。
- 如果存在正整数 i, j 使得 $1 \le i < j \le n$ 而且 $A[i] > A[j]$，则这一个有序对称为 A 的一个逆序对。逆序对的数量称作逆序数。
- 求一个数列的逆序数？ $n <= 10^5$

例题.逆序对

- 树状数组 or 平衡树，即按顺序考虑每个 Ai，在它之前有多少个比它大的数，略

- 归并排序

1.划分问题：把序列分成元素个数尽量相等的两半

2.递归求解：把两半元素分别排序

3.合并问题：把两个有序表 O(n) 合并为一个，双指针

求逆序对只要考虑两个有序表合并的时候新产生多少逆序对

搜索

- 深度搜索和广度搜索，优先选择比较好实现的深度搜索。
- 一般只在网格图寻路，分层图 dp 的时候使用广搜。
- 特别注意：不要用深度搜索求最短路！
- 做好复杂度估计，直接回溯通不过可以考虑折半搜索或者剪枝。
- 如果有必要，用 hash（推荐 map）存储已经搜过的状态。

例题.买汽水

- 暑假有 N 天，总共有 M 块钱，每天可以选择花 pi 的钱买汽水，也可以不买。问最多能花掉多少钱?

- 30%的数据，N <= 20

- 100%的数据，N <= 40, pi <= 10^9, M <= 10^9

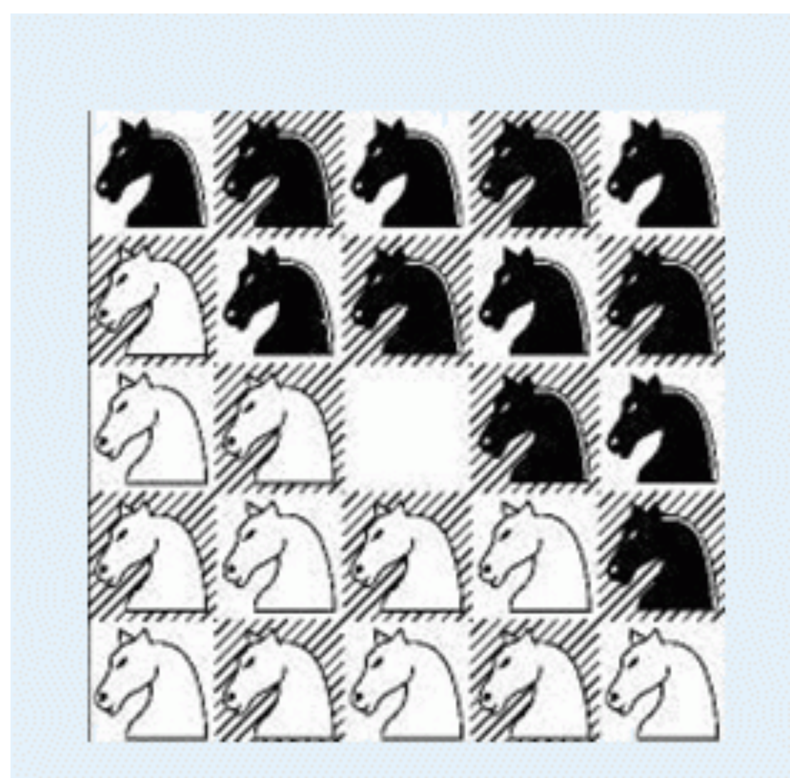
例题.买汽水

- 首先这个问题看起来很像 01 背包的 NPC，没有特殊性质的话，解法一定是搜索！
- 30%的数据，N <= 20，2^N 回溯 2^20 有 6 个 0，可以通过
- 100%的数据，N <= 40, pi <= 10^9, M <= 10^9
  - N = 40 启发我们使用折半搜索
  - 把前一半的方案 {x} 搜出来存在一个数组里排序，对于后一半的每一个方案 y，在 {x} 中二分查找 M - y 的前驱
  - 也可以用平衡树 (set) 维护

BZOJ1085. 骑士精神

- 在一个5×5的棋盘上有12个白色的骑士和12个黑色的骑士， 且有一个空位。在任何时候一个骑士都能按照骑士的走法（它可以走到和它横坐标相差为1，纵坐标相差为2或者横坐标相差为2，纵坐标相差为1的格子）移动到空位上。

- 给定一个初始的棋盘，怎样才能经过移动变成如下目标棋盘，求最少的步数（保证步数 <= 15)



BZOJ1085. 骑士精神

- 双向广搜，正着搜 7 步，反着搜 8 步
- 迭代加深搜索 + 剪枝
- A* 估计当前至少还要跳几步，如果一定得不到最优解则剪枝

练习题

CF140C.New Year Snowmen

- 要堆起一个雪人，需要三个不同大小的雪球
- 现在有 n 个雪球，第 i 个雪球大小是 ai 个，问最多能堆起多少个雪人，给定的雪球大小可能相同
- $1 \le n \le 10^5$, $1 \le ai \le 10^9$

CF140C.New Year Snowmen

- 数量少的雪球更“珍贵”, 数量最多的雪球先用
- 计算每种大小的雪球的个数，并放入堆中
- 每次取出前三多的雪球大小，用这三种雪球堆一个雪人

CF175C. Geometry Horse

- 有 n 种数字，第 i 种数字的数量为 ki，价值为 ci
- 依次取走数字，取走某个数字时获得的价值为 ci * f
- 其中 f 表示当前系数，初始为1
- 当取走的数字个数达到 pi 后，f += 1，pi 表示第 i 个素数
- 可以以任意顺序取走数字，求可获得的最大价值和
- n<=100，k_i <= 100

CF175C. Geometry Horse

- 当前系数 f 单调不降
- 越后取的数字，乘的当前系数 f 越大
- 以数字大小排序所有数字，按价值从小到大取走数字，计算答案

CF486B.OR in Matrix

- A, B 都是 n*m 的 01 矩阵，已知 B 矩阵是由 A 矩阵以一种规则生成

- Bij 是由 A 矩阵的第 i 行的所有元素和第 j 列的所有元素进行或运算 得到

- 给定 B 矩阵，求是否存在一个矩阵 A 能生成 B

- 1 ≤ n, m ≤ 100

CF486B.OR in Matrix

- 如果矩阵B的某个值为 0，则 A 矩阵整行整列都是 0
- 如果矩阵B的某个值为 1，那 A 矩阵行或列上至少有 1 个 1
- 先填 0，其余都填 1，最后验证一下

CF724B.Batch Sort

- 给一个 n * m 的矩阵，每一行是一个 1-m 的全排列
- 可以交换矩阵的两列，然后每一行还可以交换两个元素
- 问能不能使得每一行都是单调递增的
- 1 <= n, m <= 100

CF724B.Batch Sort

- 暴力枚举要交换的两列，然后逐行判断是否能通过一次交换使得一整行单调递增

CF460C.Present

- 有 n 盆花，编号 1 到 n，每盆花都有高度值 h_i
- 浇 m 次水，每次只能浇连续的 w 盆花，每浇一次，被浇的花高度值 +1。
- 希望让其中最矮的花最高，问浇完水后最矮的花的高度的最大值?
- $1 \le n, m, w \le 10^5, 1 \le h_i \le 10^9$

CF460C.Present

- 二分最优解，贪心判定
- 从左到右，如果某盆花小于二分值，将其以及后面的 w 盆花 +1
- 用线段树 / 差分 + 前缀和维护

CF482A.Diverse Permutation

- 构造 n 的一个全排列，使其相邻数之间的差值有 K 种，若不存在输出 No
- 例：1 5 4 2 3 的差值序列是 4 1 2 1，有 3 种差值
- 1 <= n, k <= 10^5

CF482A.Diverse Permutation

- 1 ～ n 最多凑出 n – 1 种差值
- 8 1 7 2 6 3 5 4 的差值序列是 7 6 5 4 3 2 1
- 构造 k 项差值为 2 到 k， 其余的差值全为 1 即可

CF425A. Sereja and Swaps

- 给一个长为 n 的序列，以及交换次数 k
- 每次可以在原先的序列中任意交换两个数
- 交换后找一个最大子串和，输出其可能的最大值。
- $1 \le n \le 200, 1 \le k \le 10$

CF425A. Sereja and Swaps

- 枚举一个子串，将子串内最小的和子串外最大的尝试交换

CF1187C.Vasya And Array

- 构造一个长为 n 的序列 A，满足 m 个限制条件
- 限制条件有两种

1. 要求 $[l_i, r_i]$ 这一段序列是严格递增的
2. 要求 $[l_i, r_i]$ 这一段序列不是递增的

- 问满足条件的序列是否存在？如果存在则输出任一合法序列。
- $1 \le n \le 1000$, $1 \le m \le 1000$，要求 $0 \le A_i \le 10^9$

CF1187C.Vasya And Array

- 考虑差分后的序列，初始值为全 0
- 如果要求一段是严格递增的，则这一段全设成 1
- 把所有严格递增的先满足，再考虑每个非递增的，在这些区间内至少找一个设成 -1

POI2005. Toy Cars

- 有 n 个不同的玩具，接下来 p 个时刻每次要玩其中的一个玩具
- 地上可以放 k 个玩具，如果一个要玩的玩具不在地上，就要到架子上去拿，地上的玩具超过 k 的时候要选择一个放回
- 求从架子上拿玩具的最小次数
- k, n <= 10^5, p <= 5 * 10^5

POI2005. Toy Cars

- 一开始肯定把前 K 个要玩的玩具放地上，考虑放回哪个玩具
- 记录下每个玩具下次出现的时间，每次要放回的话选一个离现在时刻最远的玩具
- 用堆维护这一过程

CF767E. Change Free

- 现在有面值为100的纸币和1元的硬币，纸币无限多，但是硬币只有 m 个。
- 接下来 n 天，每天都要去食堂花费 c[i] 元。
- 已知收银员在第 i 天找零 x 元的话，不满意度会增加 x * w[i]
- 求最小的不满意度之和，并输出方案（每天用几张纸币和几个硬币）
- $n \le 10^5$，$m \le 10^9$

CF767E. Change Free

- 只用考虑 c[i] % 100 的部分。
- 我们可以每天都用硬币，直到第 k 天发现硬币不够了。那就说明我们必须在 $1 \sim k$ 天中的某一天找一次零。
- 假设我们选择第 i 天（$1 \leq i \leq k$），本来是花了 $c[i]$ 个硬币。
- 现在改成使用纸币，节省下了 $c[i]$ 个硬币，还找回了（$100-c[i]$）个硬币。
- 因此无论 $c[i]$ 多少，都会多出 100 个硬币。

CF767E. Change Free

- 选择第 i 天的代价是 w[i](100-c[i])
- 因此，如果第 k 天硬币不够了，那么就在 $1\sim k$ 天中选一个代价最小的，兑换 100 个硬币即可。
- 用最小堆来维护代价。
- 这样就保证每一步决策都是最优的，贪心策略正确！

CF1190B. Tokitsukaze, CSL and Stone Game

- n 堆石子，每堆有 ai 个，两个人玩取石子游戏。
- 两人轮流，每次取一个。如果一个人没有石子可以取，或者取完石子后，存在两堆一样多的石子，则失败。
- 一堆的石子数可以是 0，不能是负数。
- 两人都遵循最优策略，问先手必胜还是后手必胜?
- 1 <= n <= 10^5, 0 <= ai <= 10^9

CF1190B. Tokitsukaze, CSL and Stone Game

- 排序不影响答案。

- 如果初始就有两堆一样的石子，先手必须取一个，如果取完以后还有两堆一样的，则后手胜利。

- 接下来可以直接计算总共能取多少次石子，即把第 i 小的石子变成 i (从 0 开始标号)。以总操作数的奇偶性判断胜负。

CF196C. Paint Tree

- 给定二维平面上的N个点，保证没有三点共线。
- 给定一棵树，请在二维平面上的 N 个点之间连 N-1 条线，使得这些线无交且与给定的树同构。
- 数据保证有解，输出任意方案。
- N<=1500

CF196C. Paint Tree

- 选择树中的任意一点作为根，用 dfs 求出每个子树的大小
- 选二维平面中最左边的点为根
- 对其他点做极角排序, 按照极角序分配给每个子树, 分治处理
- 无三点共线→必然有解

LOJ6560. 小奇取石子

- 有 n 堆石子，第 i 堆石子有 ai 个，最多取 m 堆石子，请问在要求总石子数不超过 k 的情况下最多能取多少石子。

数据分为 A、B、C 三组，各占 30%、30%、40%；

对于 A 组数据，$1 \le m \le n \le 10, 1 \le k \le 1000, 1 \le a_i \le 100$;

对于 B 组数据，$1 \le m \le n \le 20, 1 \le k \le 10^8, 1 \le a_i \le 10^6$;

对于 C 组数据，$1 \le m \le n \le 200, 1 \le k \le 2500, 1 \le a_i \le 50$;

LOJ6560. 小奇取石子

- 在n个数字当中取出m个，使得总和小于k。这是一个比较显然的背包动态规划模型，复杂度是 $O(kmn)$
- 当然，对于n较小的数据，我们甚至可以使用 $O(2^n)$ 的算法暴力搜索每一种匹配方式，然后判断是否符合条件。
- 同时，我们注意到B组数据的k范围比较大，如果采用背包dp算法时间复杂度太高，我们也开不出那么大的数组来记录状态。
- 但是B组数据的n太小了，采用暴力搜索反而能过。所以，我们对于不同组的数据需要不同处理。

LOJ6560. 小奇取石子

数据分为 A、B、C 三组，各占 30%、30%、40%；

对于 A 组数据，$1 \le m \le n \le 10, 1 \le k \le 1000, 1 \le a_i \le 100$;

对于 B 组数据，$1 \le m \le n \le 20, 1 \le k \le 10^8, 1 \le a_i \le 10^6$;

对于 C 组数据，$1 \le m \le n \le 200, 1 \le k \le 2500, 1 \le a_i \le 50$;

- 对于A组数据，我们注意到n和m的数据范围很小，只有10，那么直接暴力搜索或者做背包动态规划都能通过。
- 对于B组数据，由于k的范围过大，我们只能暴力搜索。
- 对于C组数据，由于n的范围过大，我们只能背包dp。

codechef. CLPERM

- 数字 1 - N 中丢失了 K 个数字，求：不能用剩余数字加和得到的数字中 最小的一个 的奇偶性，数字不能重复用
- 1 <= N <= 10^9, 1 <= K <= 10^6

codechef. CLPERM

- 设最小的末丢失数字是 mn
- 如果当前凑出了 1 ～ x，只要 $mn < 1 + 2 + \ldots + x$，则 1 ~ (x + mn) 都可以凑出来
- 从小到大依次考虑丢失的数字
- 复杂度 $O(\sqrt{n})$

BZOJ 2527. Meteors

有 n 个国家和 m 个空间站，每个空间站都属于一个国家，一个国家可以有多个空间站，所有空间站按照顺序排成一行。

现在，将会有 k 场流星雨降临，每一场流星雨都会给区间 [li，ri] 内的每个空间站带来 ai 单位的陨石，每个国家都有一个收集陨石的目标 pi，即第 i 个国家需要收集 pi 单位的陨石。

询问：每个国家最早完成陨石收集目标是在第几场流星雨后，如果所有流星雨结束后某个国家还未完成收集目标则输出 -1

n ≤ m ≤ 300000, ai, pi ≤ 10^9

BZOJ 2527. Meteors

如果只考虑第 i 个国家最早完成收集目标的时间，可以二分

n 个国家都满足二分的性质，考虑整体二分

二分时间 mid，将 [L, mid] 的每个修改操作执行一遍（利用树状数组）

可以看成对于时间维护了一棵二分树

对每个询问，检查是否到达了目标，达到了放左边，没达到的放右边

对于放在右边的点，要从目标里减去 [L, mid] 的修改操作的贡献

O(nlogn)