NOIP基本算法

Google 李煜东 2019/1/26

二分

- 二分法
- 整数域二分，在单调序列中查找某个值的位置
- 实数域二分
- 答案具有单调性，二分答案把求解转化为判定（复杂度理论：判定的难度小于求解）

- 三分法
- 求严格单峰函数极值

二分答案

- 通过二分，我们可以**把求最优解的问题，转化为给定一个值 mid，判定是否存在一个可行方案评分达到 mid 的问题**。

- 经典例题：
- N 本书排成一行，第 i 本厚度 a_i，
- 请把它们分成**连续的** M 组，使 T 最小化，
- 其中 T 表示厚度之和最大的一组的厚度。

最长子段

- 给定一个数组 $a[1\sim n]$，求一个尽可能长的非空连续子段，使得子段和 $\leq S$。
- $n \leq 10^5$， $S, |a_i| \leq 10^9$

最长子段

- 给定一个数组 $a[1 \sim n]$，求一个尽可能长的非空连续子段，使得子段和 $\le S$。
- $n \le 10^5$，$S, |a_i| \le 10^9$

- 二分答案
- 转化为：判定是否存在一个长度 $\ge m$ 的子段，使得子段和 $\le S$
- 设 sum 是 a 的前缀和，即 $sum[i] - sum[j] \le S$ 并且 $i - j \ge m$
- 对于每个 i，可求出 $val = \max_{0 \le j \le i-m} \{sum[j]\}$，判定是否有 $sum[i] - val \le S$
- 随着 i 的增长，每次只有一个新的 j 进入范围，用一个变量维护即可
- 整个算法时间复杂度 $O(n \log n)$

Best Cow Fences

- http://poj.org/problem?id=2018
- 题意：给定一个正整数数列 A，求一个平均数最大的长度不小于 L 的子段。

Best Cow Fences

- http://poj.org/problem?id=2018
- 题意：给定一个正整数数列 A，求一个平均数最大的长度不小于 L 的子段。

- 二分答案（平均值）
- 判定是否存在一个长度不小于 L 的子段，平均数不小于二分的值 m
- 判定方法：
- 把数列中每个数都减去平均值，问题变为是否存在长度不小于 L 的子段，子段和非负
- 转化为前缀和相减的形式 $sum[i] - \min\limits_{0 \le j \le i-L} \{sum[j]\} \ge 0$
- 与上一道题类似

双指针扫描

- Subsequence http://poj.org/problem?id=3061
- 给定序列 A 和一个整数 S，求一个长度最小的连续子段，使得子段和 $\geq S$。

双指针扫描

- Subsequence http://poj.org/problem?id=3061
- 给定序列 A 和一个整数 S，求一个长度最小的连续子段，使得子段和 $\geq S$。
- 前缀和 + 二分 $O(n \log n)$
- 双指针扫描 $O(n)$ —— 枚举左端 l，所求的子段右端 r 是单调递增的

双指针扫描

- Subsequence http://poj.org/problem?id=3061
- 给定序列 A 和一个整数 S，求一个长度最小的连续子段，使得子段和 $\geq S$。
- 前缀和 + 二分 $O(n \log n)$
- 双指针扫描 $O(n)$ —— 枚举左端 l，所求的子段右端 r 是单调递增的

- Costume Party http://poj.org/problem?id=3663
- 给定序列 A 和一个整数 S，求有多少对 (i, j) 满足 $A_i + A_j \leq S$。

双指针扫描

- Subsequence http://poj.org/problem?id=3061
- 给定序列 A 和一个整数 S，求一个长度最小的连续子段，使得子段和 $\geq S$。
- 前缀和 + 二分 $O(n\log n)$
- 双指针扫描 $O(n)$ —— 枚举左端 l，所求的子段右端 r 是单调递增的

- Costume Party http://poj.org/problem?id=3663
- 给定序列 A 和一个整数 S，求有多少对 (i,j) 满足 $A_i+A_j\leq S$。
- 把序列 A 排序
- 枚举 i，满足要求的 j 的右端位置是单调递减的

递推

- 递推通常指利用可重复的步骤不断从已知项计算未知项的过程。

- 数列的递推公式?
- 动态规划的状态转移方程?

BZOJ4300

- https://www.lydsy.com/JudgeOnline/problem.php?id=4300
- 给定一个长度为 n 的数列 a_i，求 a_i 的子序列 b_i 的最长长度，满足 $b_i \,\&\, b_{i-1} \neq 0$。
- $n \leq 200000, a_i \leq 10^9$

BZOJ4300

- 给定一个长度为 n 的数列 a_i，求 a_i 的子序列 b_i 的最长长度，满足 $b_i \,\&\, b_{i-1} \neq 0$。
- $n \leq 200000, a_i \leq 10^9$

- $\mathrm{O}(n^2)$ 递推
- $F[i]$ 表示以 i 结尾的子序列最长长度，$F[i] = \max\{F[j] + 1\}$ 其中 $a_j \,\&\, a_i \neq 0$

- $\mathrm{O}(n \log \mathrm{MAX})$ 按位递推
- $D[k]$ 表示最后一个数二进制第 k 位不为零的子序列最长长度
- 若 a_i 第 k 位不为零，$F[i] = \max\{D[k] + 1\}$，$D[k] = \max(D[k], F[i])$

皇帝的烦恼

- https://www.lydsy.com/JudgeOnline/problem.php?id=1863
- n 个人围成一个环，每个人要求有 a_i 种不同颜色的勋章，任意相邻的两个人不能有相同颜色的勋章，求最少需要多少种颜色
- $n \le 200000, a_i \le 100000$

皇帝的烦恼

- https://www.lydsy.com/JudgeOnline/problem.php?id=1863
- n 个人围成一个环，每个人要求有 a_i 种不同颜色的勋章，任意相邻的两个人不能有相同颜色的勋章，求最少需要多少种颜色
- $n \le 200000, a_i \le 100000$

- 二分答案
- 转化为：判定用 m 种颜色能否达到题目要求
- 二分下界为 $\max\{a_i + a_{i+1}\}$，上界为 $\sum a_i$

皇帝的烦恼

- https://www.lydsy.com/JudgeOnline/problem.php?id=1863
- n 个人围成一个环，每个人要求有 a_i 种不同颜色的勋章，任意相邻的两个人不能有相同颜色的勋章，求最少需要多少种颜色
- $n \le 200000, a_i \le 100000$

- 设 $F[i], G[i]$ 表示满足前 i 个人的要求时，第 i 个人和第 1 个人的最少、最多相同颜色
- $F[i] = \max\left(a_i - \left(m - a_{i-1} - (a_1 - G[i-1])\right),\ 0\right)$
- $G[i] = \min(a_i,\ a_1 - F[i-1])$
- 若 $F[n] = 0$ 则 m 是可行的

前缀和

- 对于一个给定的数列A，它的前缀和数列S是通过递推能求出的基本信息之一：
- $S[i] = \sum_{j=1}^{i} A[j]$

- 一个部分和，即数列A中某个下标区间内的数的和，可以表示为前缀和相减的形式：
- $\text{sum}(l, r) = \sum_{i=l}^{r} A[i] = S[r] - S[l-1]$

- 在二维数组（矩阵）中，也有类似的递推形式，可求出二维前缀和，进一步计算出二维部分和。

差分

- 长度为 n 的序列 A 的差分序列是一个长度为 n 的序列 B
- 其中 $B_1 = A_1$，$B_i = A_i - A_{i-1}\ (2 \le i \le n)$

- 差分序列 B 的前缀和数组就是原序列 A
- 把 A 的区间 $[l, r]$ 加 d，B 的变化为：B_l 加 d，B_{r+1} 减 d

BZOJ1218 激光炸弹

- http://www.lydsy.com/JudgeOnline/problem.php?id=1218

BZOJ1218 激光炸弹

- http://www.lydsy.com/JudgeOnline/problem.php?id=1218

- 由于 X_i, Y_i 的值在0~5000之间，所以我们可以建立一个二维数组 A，其中 $A[i,j]$ 就等于位置 (i,j) 上的所有目标的价值之和。即对于每个目标，令 $A[X_i, Y_i] += W_i$。
- 接下来我们求出 A 的二维前缀和 S，即：$S[i,j] = \sum_{x=1}^{i} \sum_{y=1}^{j} A[x,y]$
- 事实上，有更快的递推式 $S[i,j] = S[i-1,j] + S[i,j-1] - S[i-1,j-1] + A[i,j]$
- 枚举正方形右下角坐标，用下式更新答案：
- $\sum_{x=i-R+1}^{i} \sum_{y=j-R+1}^{j} A[x,y] = S[i,j] - S[i-R,j] - S[i,j-R] + S[i-R,j-R]$

IncDec Sequence

- https://www.lydsy.com/JudgeOnline/problem.php?id=3043
- 给定一个长度为 n 的数列 $\{a_1, a_2, \dots, a_n\}$，每次可以选择一个区间 $[l, r]$，使这个区间内的数都加一或者都减一。
- 问至少需要多少次操作才能使数列中的所有数都一样，并求出在保证最少次数的前提下，最终得到的数列有多少种。
- $n \le 100000$

IncDec Sequence

- https://www.lydsy.com/JudgeOnline/problem.php?id=3043

- 求出 a 的差分序列 b
- 相当于每次可以选出 b 中任意两个数（也可以选 b_{n+1}），一个 +1，一个 -1
- 目标是把 $b[2\sim n]$ 变为全零

- 设 $b[2\sim n]$ 中正数总和为 p，负数总和为 q
- 正负数配对，最少操作次数为 $\max(p,q)$
- 剩余 $|p-q|$ 个未配对，每个可以选与 b_1 或 b_{n+1} 配对，共 $|p-q|+1$ 种选法

Tallest Cow

- http://poj.org/problem?id=3263
- 有 N 头牛站成一行。两头牛能够相互看见，当且仅当它们中间的牛身高都比它们矮。
- 我们只知道其中最高的牛是第 P 头，它的身高是 H，不知道剩余 $N-1$ 头牛的身高。
- 我们还知道 M 对关系，每对关系都指明了某两头牛 A_i 和 B_i 可以相互看见。
- 求每头牛的身高最大可能是多少。
- $1 \le N, M \le 10^4$，$1 \le H \le 10^6$。

Tallest Cow

- http://poj.org/problem?id=3263
- 建立一个数组 C，数组中起初全为0。
- 若一条关系指明 A_i 和 B_i 可以互相看见（不妨设$A_i < B_i$），则把数组 C 中下标为 $A_i + 1$ 到 $B_i - 1$ 的数都减去1，意思是在 A_i 和 B_i 中间的牛，身高至少要比它们小1。
- 由于第P头牛是最高的，所以最终 $C[P]$ 一定为0，第 i 头牛的身高就等于 $H + C[i]$。

- 可转化为建立一个数组 D，对于每对 A_i 和 B_i，令 $D[A_i + 1]$ 减去1，$D[B_i]$ 加上1。
- 其含义是："身高减小1"的影响从 $A_i + 1$ 开始，持续到 $B_i - 1$，在 B_i 结束。
- 最后，C 就等于 D 的前缀和，即 $C[i] = \sum_{j=1}^{i} D[j]$。

排序

- 选择排序 $O(n^2)$
- 插入排序 $O(n^2)$
- 冒泡排序 $O(n^2)$
- 计数排序 $O(n+m)$
- 基数排序 $O(n \log m)$
- 桶排序 $O(n)\sim O(n^2)$
- 归并排序 $O(n \log n)$
- 堆排序 $O(n \log n)$
- 快速排序 $O(n \log n)$

工作安排

- 有 n 件工作，第 i 件需要占用 $[a_i, b_i]$ 的时间，不能同时做多件工作，求最多能完成几件
- $n \leq 100000$，$1 \leq a_i, b_i \leq 10^9$

工作安排

- 有 n 件工作，第 i 件需要占用 $[a_i, b_i]$ 的时间，不能同时做多件工作，求最多能完成几件
- $n \leq 100000$，$1 \leq a_i, b_i \leq 10^9$

- 解法一：排序 + 动态规划 + 二分，$O(n \log n)$
- 按右端点排序，$F[i]$ 表示前 i 件工作最多完成几件
- $F[i] = \max(F[i-1], F[j] + 1)$，其中 j 是最后一个满足 $b_j < a_i$ 的位置，可二分确定
- 解法二：排序 + 贪心扫描，$O(n \log n)$
- 按左端点排序，扫描过程中维护变量 p
- 若 $a_i > p$，答案新增一件工作，否则 $p = \min(p, b_i)$

冒泡排序相关

- 交换次数 = 逆序对数
- 比较次数 = 外层循环遍数 * $(n-1)$
- 遍数 = max{ 每个数前面有几个比它大 }

Ultra-QuickSort

- http://poj.org/problem?id=2299
- 给定一个长度为 $n(n \le 500000)$ 的序列 a，如果只允许进行比较和交换相邻两个数的操作，求至少需要多少次交换才能把 a 从小到大排序。

Ultra-QuickSort

- http://poj.org/problem?id=2299
- 给定一个长度为 $n(n \leq 500000)$ 的序列 a，如果只允许进行比较和交换相邻两个数的操作，求至少需要多少次交换才能把 a 从小到大排序。

- 只通过比较和交换相邻两个数值的排序方法，实际上就是冒泡排序算法。
- 在排序过程中每找到一对大小颠倒的相邻数值，把它们交换，就会使整个序列的逆序对个数减少 1。
- 最终排好序后逆序对个数显然为 0。
- 所以对 a 进行冒泡排序需要的最少交换次数就是序列 a 中逆序对的个数。

Out of Sorts [Gold]

- https://www.lydsy.com/JudgeOnline/problem.php?id=5278

Out of Sorts [Gold]

- https://www.lydsy.com/JudgeOnline/problem.php?id=5278
- 先离散化
- 双向冒泡排序遍数 = max{前 i 个位置上 > i的数有多少个}

- 因为对于每个 i
- 向后扫会保证把前 i 个位置上一个 > i 的数扔到后边去
- 向前扫会保证被换到前 i 个位置上的新数是 ≤ i 的

- 扫一遍序列，权值树状数组维护后缀和即可

Out of Sorts [Platinum]

- https://www.lydsy.com/JudgeOnline/problem.php?id=5277

Out of Sorts [Platinum]

- https://www.lydsy.com/JudgeOnline/problem.php?id=5277
- 先离散化
- 如果 i 不在前 i 个位置上，那么一次冒泡必然会把 i 向前移动恰好 1 个位置
- 满足 “前 i 个位置是 $1\sim i$ 这些数” 所需的时间是 $t[i] = \max\{pos[1..i]\} - i$

- 第 i 个数对答案产生的的贡献为 $\max(t[i-1], t[i])$
- 此时 i 左右两侧都有分割点，长度为 1 停止递归

幻化

- 给定长度为 n 的整数序列 a 和 b，两个序列均是 $[1, n]$ 的一个排列
- 通过若干次交换 $a[i], a[j]$ 使序列 a 变成 b，交换的代价是 $|j - i|$
- 求最小代价

幻化

- 给定长度为 n 的整数序列 a 和 b，两个序列均是 $[1,n]$ 的一个排列
- 通过若干次交换 $a[i], a[j]$ 使序列 a 变成 b，交换的代价是 $|j-i|$
- 求最小代价

- 答案为每个数在 a 中的位置和 b 中的位置的距离之和 / 2
- 只需证明每次一定可以找到一对 (a_i, a_j)，交换以后二者到目标位置的距离都变近了
- 不妨设 a_r 是最后一个位置不对的数，a_l 是应该换到 r 位置的数
- 若 a_r 的目标位置比 l 靠前，交换 (a_l, a_r) 即可
- 否则，根据抽屉原理，一定存在 $a_p (l < p < r)$ 目标位置比 l 靠前，交换 (a_l, a_p) 即可

中位数

- 货仓选址模型
- 在一条数轴上有 N 家商店，它们的坐标分别为 $A[1\sim N]$。现在需要在数轴上建立一家货仓，每天清晨，从货仓到每家商店都要运送一车商品。为了提高效率，求把货仓建在何处，可以使得货仓到每家商店的距离之和最小。
- 建在 A 排序后的中位数 $A\left[\frac{1+n}{2}\right]$ 处

中位数

- 货仓选址模型
- 在一条数轴上有 N 家商店，它们的坐标分别为 $A[1 \sim N]$。现在需要在数轴上建立一家货仓，每天清晨，从货仓到每家商店都要运送一车商品。为了提高效率，求把货仓建在何处，可以使得货仓到每家商店的距离之和最小。
- 建在 A 排序后的中位数 $A\left[\frac{1+n}{2}\right]$ 处

- 环形均分纸牌模型
- $B[i] = A[i] -$ 平均数
- S 为 B 的前缀和，把 S 排序，取中位数 $S\left[\frac{1+n}{2}\right]$，答案为 $\sum\left|S[i] - S\left[\frac{1+n}{2}\right]\right|$

贪心

- 贪心是一种在每次决策时采取当前意义下最优策略的算法。
- 使用贪心法，要求局部最优性能够导出问题整体最优性。

- 贪心法通常需要证明，常见的证明手段：
- 微扰（邻项交换）、范围缩放
- 决策包容性
- 数学归纳法、反证法

叠罗汉

- 有 n 个罗汉，第 i 个罗汉的重量为 a_i，托举力量为 b_i
- 求最多能选出多少个罗汉，使得他们按照某种方式叠起来以后，前 $i-1$ 个罗汉的总重量不超过第 i 个罗汉的托举力量
- $n \le 100000$，类似题目 http://poj.org/problem?id=3045

叠罗汉

- 有 n 个罗汉，第 i 个罗汉的重量为 a_i，托举力量为 b_i
- 求最多能选出多少个罗汉，使得他们按照某种方式叠起来以后，前 $i-1$ 个罗汉的总重量不超过第 i 个罗汉的托举力量
- $n \le 100000$，类似题目 http://poj.org/problem?id=3045

- 按照 $a_i + b_i$ 排序
- 依次尝试每个罗汉，如果能叠到最底下（b_i 足够大），答案 +1
- 否则用 a_i 替换掉当前参与叠罗汉的最重的一个（前提是他比 a_i 重）
- 可以用二叉堆维护

国王游戏

- NOIP2012
- 恰逢 H 国国庆，国王邀请 n 位大臣来玩一个有奖游戏。首先，他让每个大臣在左、右手上面分别写下一个整数，国王自己也在左、右手上各写一个整数。然后让这 n 位大臣排成一排，国王站在队伍的最前面。
- 排好队后，所有的大臣都会获得国王奖赏的若干金币，每位大臣获得的金币数分别是：排在该大臣前面的所有人的左手上的数的乘积除以他自己右手上的数，然后向下取整得到的结果。国王不希望某一个大臣获得特别多的奖赏，所以他想请你帮他重新安排一下队伍的顺序，使得获得奖赏最多的大臣，所获奖赏尽可能的少。

国王游戏

- NOIP2012
- 恰逢 H 国国庆，国王邀请 n 位大臣来玩一个有奖游戏。首先，他让每个大臣在左、右手上面分别写下一个整数，国王自己也在左、右手上各写一个整数。然后让这 n 位大臣排成一排，国王站在队伍的最前面。
- 排好队后，所有的大臣都会获得国王奖赏的若干金币，每位大臣获得的金币数分别是：排在该大臣前面的所有人的左手上的数的乘积除以他自己右手上的数，然后向下取整得到的结果。国王不希望某一个大臣获得特别多的奖赏，所以他想请你帮他重新安排一下队伍的顺序，使得获得奖赏最多的大臣，所获奖赏尽可能的少。

- 按照左右手上的数的乘积升序排序，就是最优排队方案。
- 证明方法：邻项交换

Commando War

- UVA11729
- 有 n 个任务，第 i 个任务需要 A_i 分钟写好代码，然后代码会自己跑 B_i 分钟做完任务。
- 求一种写代码的顺序，使得所有任务尽早被做完。不能同时写两个任务的代码。

Commando War

- UVA11729
- 有 n 个任务，第 i 个任务需要 A_i 分钟写好代码，然后代码会自己跑 B_i 分钟做完任务。
- 求一种写代码的顺序，使得所有任务尽早被做完。不能同时写两个任务的代码。

- 按照 B_i 从大到小的顺序交代任务即可。
- 证明方法：邻项交换

DZY Loves Physics

- http://codeforces.com/problemset/problem/444/A
- 给定 n 个点 m 条边的无向图 $(1 \le n \le 500)$
- 求一个点数 ≥ 2 的连通的导出子图，使得点权和除以边权和最大

DZY Loves Physics

- http://codeforces.com/problemset/problem/444/A
- 给定 n 个点 m 条边的无向图 $(1 \le n \le 500)$
- 求一个点数 ≥ 2 的连通的导出子图，使得点权和除以边权和最大

- 比较 $\frac{x+y}{p}, \frac{y+z}{q}$ 和 $\frac{x+y+z}{p+q}$
- $\frac{x+y+z}{p+q}$ 不可能是三者中的最大值
- 故所求的子图必然由两个点和一条边构成

Radar Installation

- http://poj.org/problem?id=1328
- 校长想通过监控设备覆盖学校内的 N 座建筑物
- 每座建筑物被视作一个质点，在笛卡尔坐标系中给出它们的坐标 (x, y)，并且所有建筑物均处在 x 轴的上方
- 由于学校的供电和传输线路均沿 x 轴，所以监控设备只被允许建立在 x 轴上
- 每台监控设备的监控范围均为一个半径为 R 的圆形，圆心即为这台设备
- 现给出 N 座建筑物的坐标，问最少需要几台设备可以实现对所有建筑物的监控?

Radar Installation

- http://poj.org/problem?id=1328
- 把 x 轴上方的每个点转化为 x 轴上的一段能管辖它的区间 $l[i]$~$r[i]$。
- 按照每个区间左端点的 x 坐标从小到大排序。
- 贪心，假如当前区间 i 的左端点坐标大于已经放置的任何一台设备，则新增一台。
- 否则就让上一台已安放的设备来管辖它，安放位置与 $r[i]$ 取 min。
- 以此类推直至所有区间被管辖，输出设备个数即可。

Buy Low Sell High

- http://codeforces.com/contest/867/problem/E
- 已知 N 天的股票价格，每天可以买入一股 or 卖出一股 or 什么也不做
- N 天后必须清仓，求最多能赚多少钱

Buy Low Sell High

- http://codeforces.com/contest/867/problem/E
- 已知 N 天的股票价格，每天可以买入一股 or 卖出一股 or 什么也不做
- N 天后必须清仓，求最多能赚多少钱

- 贪心， $O(N\log N)$
- 从前往后扫描每一天的股价，维护一个小根堆
- 若堆中所有值都比当前股价高，则当前股价入堆
- 否则从堆顶那天买入（取出堆顶），从当前这天卖出（累加答案），并把当前这天的股价**入堆 2 次**

谢谢大家

lydrainbowcat@pku.edu.cn