分治法的应用

Claris

Hangzhou Dianzi University

2018 年 2 月 6 日

Overview

- 本节课主要介绍以下两种方法：

Overview

- 本节课主要介绍以下两种方法：
- 二分查找。

Overview

- 本节课主要介绍以下两种方法：
- 二分查找。
- 分治法。

二分查找

- 对于在给定区间中单调的函数，可以采用二分查找来加速查找。

二分查找

- 对于在给定区间中单调的函数，可以采用二分查找来加速查找。
- 一般的方法是，假设当前答案位于 $[l, r]$ 中，取 $mid = \frac{l+r}{2}$。

二分查找

- 对于在给定区间中单调的函数，可以采用二分查找来加速查找。
- 一般的方法是，假设当前答案位于 $[l, r]$ 中，取 $mid = \frac{l+r}{2}$。
- 根据 $f(mid)$ 决定往 $[l, mid)$ 还是 $(mid, r]$ 继续查找。

二分查找

- 对于在给定区间中单调的函数，可以采用二分查找来加速查找。
- 一般的方法是，假设当前答案位于 $[l, r]$ 中，取 $mid = \frac{l+r}{2}$。
- 根据 $f(mid)$ 决定往 $[l, mid)$ 还是 $(mid, r]$ 继续查找。
- 查找次数 $O(\log(r - l))$。

Need for Speed

有一辆速度表不准的车，当表盘读数为 s 时，真实车速为 $s + c$，c 为任意实数常量，且可能是负数。

Need for Speed

有一辆速度表不准的车，当表盘读数为 s 时，真实车速为 $s+c$，c 为任意实数常量，且可能是负数。

这辆车依次通过了 n 段道路，总共花费了 t 的时间，其中第 i 段道路的长度为 d_i，表盘读数一直为 s_i。

Need for Speed

有一辆速度表不准的车，当表盘读数为 s 时，真实车速为 $s+c$，c 为任意实数常量，且可能是负数。

这辆车依次通过了 n 段道路，总共花费了 t 的时间，其中第 i 段道路的长度为 d_i，表盘读数一直为 s_i。

已知这些信息，请求出实数常量 c。

Need for Speed

有一辆速度表不准的车，当表盘读数为 s 时，真实车速为 $s+c$，c 为任意实数常量，且可能是负数。

这辆车依次通过了 n 段道路，总共花费了 t 的时间，其中第 i 段道路的长度为 d_i，表盘读数一直为 s_i。

已知这些信息，请求出实数常量 c。

- $1 \leq n \leq 1000$。

Need for Speed

有一辆速度表不准的车，当表盘读数为 s 时，真实车速为 $s+c$，c 为任意实数常量，且可能是负数。

这辆车依次通过了 n 段道路，总共花费了 t 的时间，其中第 i 段道路的长度为 d_i，表盘读数一直为 s_i。

已知这些信息，请求出实数常量 c。

- $1 \leq n \leq 1000$。
- $1 \leq t \leq 10^6$。

Need for Speed

有一辆速度表不准的车，当表盘读数为 s 时，真实车速为 $s+c$，c 为任意实数常量，且可能是负数。

这辆车依次通过了 n 段道路，总共花费了 t 的时间，其中第 i 段道路的长度为 d_i，表盘读数一直为 s_i。

已知这些信息，请求出实数常量 c。

- $1 \le n \le 1000$。
- $1 \le t \le 10^6$。
- $1 \le d_i \le 1000$。

Need for Speed

有一辆速度表不准的车，当表盘读数为 s 时，真实车速为 $s+c$，c 为任意实数常量，且可能是负数。

这辆车依次通过了 n 段道路，总共花费了 t 的时间，其中第 i 段道路的长度为 d_i，表盘读数一直为 s_i。

已知这些信息，请求出实数常量 c。

- $1 \leq n \leq 1000$。
- $1 \leq t \leq 10^6$。
- $1 \leq d_i \leq 1000$。
- $|s_i| \leq 1000$。

Need for Speed

有一辆速度表不准的车，当表盘读数为 s 时，真实车速为 $s+c$，c 为任意实数常量，且可能是负数。

这辆车依次通过了 n 段道路，总共花费了 t 的时间，其中第 i 段道路的长度为 d_i，表盘读数一直为 s_i。

已知这些信息，请求出实数常量 c。

- $1 \le n \le 1000$。
- $1 \le t \le 10^6$。
- $1 \le d_i \le 1000$。
- $|s_i| \le 1000$。
- Source：WF 2017

Solution

- 假设已知 c，那么总时间 T 为 $\sum_{i=1}^{n} \frac{d_i}{s_i+c}$。

Solution

- 假设已知 c，那么总时间 T 为 $\sum_{i=1}^{n} \frac{d_i}{s_i+c}$。
- 随着 c 的增大，T 只会越来越小；反之随着 c 的减小，T 只会越来越大。

Solution

- 假设已知 c，那么总时间 T 为 $\sum_{i=1}^{n} \frac{d_i}{s_i+c}$。
- 随着 c 的增大，T 只会越来越小；反之随着 c 的减小，T 只会越来越大。
- 二分查找 c，使得 $T = t$ 即可。

Solution

- 假设已知 c，那么总时间 T 为 $\sum_{i=1}^{n} \frac{d_i}{s_i+c}$。
- 随着 c 的增大，T 只会越来越小；反之随着 c 的减小，T 只会越来越大。
- 二分查找 c，使得 $T = t$ 即可。
- 注意二分的上下界。

Solution

- 假设已知 c，那么总时间 T 为 $\sum_{i=1}^{n} \frac{d_i}{s_i+c}$。
- 随着 c 的增大，T 只会越来越小；反之随着 c 的减小，T 只会越来越大。
- 二分查找 c，使得 $T = t$ 即可。
- 注意二分的上下界。
- 时间复杂度 $O(n \log w)$。

Center of List of Sums

给定两个长度均为 n 的数组 a 和 b。

Center of List of Sums

给定两个长度均为 n 的数组 a 和 b。

将所有 n^2 对 $a_i + b_j$ 从小到大排序，从小到大依次输出第 $\frac{n(n-1)}{2} + 1$ 到第 $\frac{n(n+1)}{2}$ 项。

Center of List of Sums

给定两个长度均为 n 的数组 a 和 b。

将所有 n^2 对 $a_i + b_j$ 从小到大排序，从小到大依次输出第 $\frac{n(n-1)}{2} + 1$ 到第 $\frac{n(n+1)}{2}$ 项。

- $1 \leq n \leq 200000$。

Center of List of Sums

给定两个长度均为 n 的数组 a 和 b。

将所有 n^2 对 $a_i + b_j$ 从小到大排序，从小到大依次输出第 $\frac{n(n-1)}{2} + 1$ 到第 $\frac{n(n+1)}{2}$ 项。

- $1 \leq n \leq 200000$。
- $0 \leq a_i, b_i \leq 10^9$

Center of List of Sums

给定两个长度均为 n 的数组 a 和 b。

将所有 n^2 对 $a_i + b_j$ 从小到大排序，从小到大依次输出第 $\frac{n(n-1)}{2} + 1$ 到第 $\frac{n(n+1)}{2}$ 项。

- $1 \le n \le 200000$。
- $0 \le a_i, b_i \le 10^9$
- Source：XVIII Open Cup named after E.V. Pankratiev. Grand Prix of SPb

Dirt Ratio

给定一个长度为 n 的序列，请找到一段非空连续区间 $[l, r]$，使得不同元素的个数除以区间长度最小。

Dirt Ratio

给定一个长度为 n 的序列，请找到一段非空连续区间 $[l, r]$，使得不同元素的个数除以区间长度最小。

- $1 \leq n \leq 60000$。

Dirt Ratio

给定一个长度为 n 的序列，请找到一段非空连续区间 $[l, r]$，使得不同元素的个数除以区间长度最小。

- $1 \leq n \leq 60000$。
- Source：2017 Multi-University Training Contest 4

分治法

- 分治法是将一个大的难以解决的问题分解成若干小的子问题。

分治法

- 分治法是将一个大的难以解决的问题分解成若干小的子问题。
- 将小的子问题逐一击破后，再将它们的结果合并。

快速幂

求 $a^b \bmod P$，$0 \leq b \leq 10^{18}$。

快速幂

求 $a^b \bmod P$，$0 \leq b \leq 10^{18}$。

- 若 $b = 0$，那么结果显然是 1。

快速幂

求 $a^b \bmod P$，$0 \le b \le 10^{18}$。

- 若 $b = 0$，那么结果显然是 1。
- 若 b 是奇数，则 $a^b = a^{b-1} \times a$。

快速幂

求 $a^b \bmod P$，$0 \le b \le 10^{18}$。

- 若 $b = 0$，那么结果显然是 1。
- 若 b 是奇数，则 $a^b = a^{b-1} \times a$。
- 若 b 是偶数，则 $a^b = (a^{\frac{b}{2}})^2$。

快速幂

求 $a^b \bmod P$，$0 \le b \le 10^{18}$。

- 若 $b = 0$，那么结果显然是 1。
- 若 b 是奇数，则 $a^b = a^{b-1} \times a$。
- 若 b 是偶数，则 $a^b = (a^{\frac{b}{2}})^2$。
- 时间复杂度 $O(\log b)$。

归并排序

将一个长度为 n 的数组 a 从小到大排序。

归并排序

将一个长度为 n 的数组 a 从小到大排序。

- 若 $n = 1$ ，那么不需要排序。

归并排序

将一个长度为 n 的数组 a 从小到大排序。

- 若 $n = 1$，那么不需要排序。
- 将当前数组平均划分为两部分 a 和 b，分别排序 a 和 b。

归并排序

将一个长度为 n 的数组 a 从小到大排序。

- 若 $n = 1$，那么不需要排序。
- 将当前数组平均划分为两部分 a 和 b，分别排序 a 和 b。
- 再将 a 和 b 归并即可。

归并排序

将一个长度为 n 的数组 a 从小到大排序。

- 若 $n = 1$，那么不需要排序。
- 将当前数组平均划分为两部分 a 和 b，分别排序 a 和 b。
- 再将 a 和 b 归并即可。
- 可以同时求出逆序对的个数。

归并排序

将一个长度为 n 的数组 a 从小到大排序。

- 若 $n = 1$，那么不需要排序。
- 将当前数组平均划分为两部分 a 和 b，分别排序 a 和 b。
- 再将 a 和 b 归并即可。
- 可以同时求出逆序对的个数。
- 时间复杂度 $T(n) = 2T(\frac{n}{2}) + O(n) = O(n \log n)$。

完全背包

有 $n+1$ 种物品，体积分别为 $0, 1, \ldots, n$，价值分别为 $w_0, w_1, \ldots, w_n$，每种物品的数量都是无限的。

完全背包

有 $n+1$ 种物品，体积分别为 $0, 1, \ldots, n$，价值分别为 $w_0, w_1, \ldots, w_n$，每种物品的数量都是无限的。

给出 m 个询问，每次给定 k 和 v，你需要选择恰好 k 个物品，使得总体积恰好为 v，且总价值最大。

完全背包

有 $n+1$ 种物品，体积分别为 $0, 1, \ldots, n$，价值分别为 $w_0, w_1, \ldots, w_n$，每种物品的数量都是无限的。

给出 m 个询问，每次给定 k 和 v，你需要选择恰好 k 个物品，使得总体积恰好为 v，且总价值最大。

- $1 \leq n \leq 1500$。

完全背包

有 $n+1$ 种物品，体积分别为 $0,1,\ldots,n$，价值分别为 $w_0,w_1,\ldots,w_n$，每种物品的数量都是无限的。

给出 m 个询问，每次给定 k 和 v，你需要选择恰好 k 个物品，使得总体积恰好为 v，且总价值最大。

- $1 \le n \le 1500$。
- $1 \le m \le 10000$。

完全背包

有 $n+1$ 种物品，体积分别为 $0, 1, \ldots, n$，价值分别为 $w_0, w_1, \ldots, w_n$，每种物品的数量都是无限的。

给出 m 个询问，每次给定 k 和 v，你需要选择恰好 k 个物品，使得总体积恰好为 v，且总价值最大。

- $1 \le n \le 1500$。
- $1 \le m \le 10000$。
- $k, v \le n$。

Non-boring sequences

一个序列被称为是不无聊的，仅当它的每个连续子序列存在一个独一无二的数字，即每个子序列里至少存在一个数字只出现一次。

Non-boring sequences

一个序列被称为是不无聊的，仅当它的每个连续子序列存在一个独一无二的数字，即每个子序列里至少存在一个数字只出现一次。

给定一个长度为 n 的整数序列，请你判断它是不是不无聊的。

Non-boring sequences

一个序列被称为是不无聊的，仅当它的每个连续子序列存在一个独一无二的数字，即每个子序列里至少存在一个数字只出现一次。

给定一个长度为 n 的整数序列，请你判断它是不是不无聊的。

- $1 \leq n \leq 200000$。

Non-boring sequences

一个序列被称为是不无聊的，仅当它的每个连续子序列存在一个独一无二的数字，即每个子序列里至少存在一个数字只出现一次。

给定一个长度为 n 的整数序列，请你判断它是不是不无聊的。

- $1 \leq n \leq 200000$。
- Source： CERC 2012

Be Friends

n 个点的无向完全图，第 i 个点的点权为 val_i，i 和 j 之间的边权为 $val_i \oplus val_j$，其中 $\oplus$ 表示二进制下按位异或。

Be Friends

n 个点的无向完全图，第 i 个点的点权为 val_i，i 和 j 之间的边权为 $val_i \oplus val_j$，其中 $\oplus$ 表示二进制下按位异或。

请计算这个图的最小生成树的边权之和。

Be Friends

n 个点的无向完全图，第 i 个点的点权为 val_i，i 和 j 之间的边权为 $val_i \oplus val_j$，其中 $\oplus$ 表示二进制下按位异或。

请计算这个图的最小生成树的边权之和。

- $1 \leq n \leq 100000$。

Be Friends

n 个点的无向完全图，第 i 个点的点权为 val_i，i 和 j 之间的边权为 $val_i \oplus val_j$，其中 $\oplus$ 表示二进制下按位异或。

请计算这个图的最小生成树的边权之和。

- $1 \leq n \leq 100000$。
- $0 \leq val_i \leq 10^9$。

Be Friends

n 个点的无向完全图，第 i 个点的点权为 val_i，i 和 j 之间的边权为 $val_i \oplus val_j$，其中 $\oplus$ 表示二进制下按位异或。

请计算这个图的最小生成树的边权之和。

- $1 \leq n \leq 100000$。
- $0 \leq val_i \leq 10^9$。
- Source： Moscow Pre-Finals Workshop 2016. National Taiwan U Selection

题目提交

课上例题：

http://acm.hdu.edu.cn/diy/contest_show.php?cid=33131

课后习题：

http://acm.hdu.edu.cn/diy/contest_show.php?cid=33132

密码：

G*&GSF&*t387tr

Thank you!