

CSP-S 数据结构提升与应用

SRwudi

October 5, 2019

开篇

什么是数据结构？

数据结构是计算机存储、组织数据的方式。

抽象数据结构的具体实现。

通过数据结构的存储和应用，可以提高计算机运作的效率。

开篇

什么是数据结构？

数据结构是计算机存储、组织数据的方式。

抽象数据结构的具体实现。

通过数据结构的存储和应用，可以提高计算机运作的效率。

在 OI 的运用当中，数据结构题通常表现什么方面的性质。

大规模输入。

有修改和查询操作。

或者是一些隐藏的不容易看出来的数据结构题需要你去维护。

开篇

本次讲课主要涉及 CSP-S 范围内的各种类型的数据结构。

基础知识和具体实现以复习和理解为主，主要主攻的应用题怎么做。

数据结构的具体实现或者代表性的基础题目不会过多赘述，本篇主讲例题都是省选难度 -NOI 难度且具有一定思维挑战的题目。

二叉堆

二叉堆

维护一系列数据中的最小值/最大值

加入： $O(\log n)$

删除： $O(\log n)$

查询： $O(1)$

实现：stl/pbds

tips: 前 k 大, 最大最小。

如何使用 stl 的 `priority_queue` 实现方便的删除。

如何使用 stl 的 `priority_queue` 实现方便的删除。

Tips: 再开一个堆表示删除堆，存储我要删除的元素，然后每当原堆顶元素和当前删除堆顶元素相同后，同时删除。

中位数

维护一个数的集合：

操作 1：插入一个数 x

操作 2：询问这个集合的中位数。

序列合并

有两个长度都是 N 的序列 A, B , 在 A 和 B 中各取一个数相加可以得到 N^2 个和,
求其中最小的 M 个这样的两个数的和
 $N, M \leq 100000$

镇静剂

有一天，病房里来了 n 位紧急需要紧急救治的病人，而你手上有 V 毫升镇静剂。

对于第 i 位病人，他需要治疗的紧急程度为 a_i ，让他镇静下来所需的剂量为 b_i 。

你可以选择一些病人，把你现在手头的镇静剂给他们注射，让他们安静下来。

如果你选择了 k 个病人： $c_1, c_2, c_3, \dots, c_k$ ，那么第 j 个病人 c_j 会得到

$$\frac{a_{c_j}}{\sum_{i=1}^k a_{c_i}} \times V.$$

毫升镇静剂，如果这个值不小于 b_{c_j} ，那么他就会镇静下来。

NOI2010 超级钢琴

题目大意

给一段数有正有负的序列，数 L, R ，求所有前 K 大长度介于 $[L, R]$ 的区间的和。

$N, K \leq 500000$

NOI2010 超级钢琴

用三元组 (l, r_1, r_2) 表示代表左端点为 l , 右端点在 $[r_1, r_2]$ 之内的区间总和。

NOI2010 超级钢琴

用三元组 (l, r_1, r_2) 表示代表左端点为 l , 右端点在 $[r_1, r_2]$ 之内的区间总和。

以左端点为 l , 右端点在 $[r_1, r_2]$ 之内的最大和区间 $[l, r]$ 作为 (l, r_1, r_2) 的键值维护堆。

NOI2010 超级钢琴

用三元组 (l, r_1, r_2) 表示代表左端点为 l , 右端点在 $[r_1, r_2]$ 之内的区间总和。

以左端点为 l , 右端点在 $[r_1, r_2]$ 之内的最大和区间 $[l, r]$ 作为 (l, r_1, r_2) 的键值维护堆。

核心思路（贪心 + 扩展）：每次取出一个堆顶的元素 (l, r_1, r_2) ，用其最大和区间更新答案，并弹出该元素。最后加入两个新元素 $(l, r_1, r-1), (l, r+1, r_2)$ 。做 K 次结束。

NOI2010 超级钢琴

用三元组 (l, r_1, r_2) 表示代表左端点为 l , 右端点在 $[r_1, r_2]$ 之内的区间总和。

以左端点为 l , 右端点在 $[r_1, r_2]$ 之内的最大和区间 $[l, r]$ 作为 (l, r_1, r_2) 的键值维护堆。

核心思路（贪心 + 扩展）：每次取出一个堆顶的元素 (l, r_1, r_2) ，用其最大和区间更新答案，并弹出该元素。最后加入两个新元素 $(l, r_1, r-1), (l, r+1, r_2)$ 。做 K 次结束。

如何寻找最大和区间：前缀最大和 ST 表。

NOI2010 超级钢琴

用三元组 (l, r_1, r_2) 表示代表左端点为 l , 右端点在 $[r_1, r_2]$ 之内的区间总和。

以左端点为 l , 右端点在 $[r_1, r_2]$ 之内的最大和区间 $[l, r]$ 作为 (l, r_1, r_2) 的键值维护堆。

核心思路 (贪心 + 扩展): 每次取出一个堆顶的元素 (l, r_1, r_2) , 用其最大和区间更新答案, 并弹出该元素。最后加入两个新元素 $(l, r_1, r-1), (l, r+1, r_2)$. 做 K 次结束。

如何寻找最大和区间: 前缀最大和 ST 表。

堆的初始值: 所有的 $(i, i+L-1, i+R-1)$ 。

二叉堆

维护一张图中两个点的连通性

支持两个操作：

加边： $O(\log n)$

查询两个点的连通性： $O(\log n)$

路径压缩： $O(\log n)$

按秩合并： $O(\log n)$

两个都用： $O(\alpha n)$

可撤销并查集

加边，撤销上次加边，维护连通性
就单独按秩合并就好，然后再开一个栈表示之前做过的操作。

例题 1

支持两个操作：

1. 加边

2. 询问 $\sum_{i=1}^n \sum_{j=i+1}^n L(i, j)$.

$L(i, j)$ 表示两个点最早什么时候连通。不连通为 0。

例题 2

一开始加一些边。最后给若干个询问，求两个点是在什么时候连通的。

例题 3

给一个数组 $a[1\dots n]$ ，开始全是 0，每次有两种操作：
 令 $a[x] = 1$ 。
 输出 $a[x\dots n]$ 中第一个 0 的位置。

Exclusive-OR

有 n 个 0/1 变量，每次告诉你 $x_i = 0/1$ 或者告诉你 $x_i \text{ xor } x_j = 1$ 这么一个关系，最后询问你一个数等于多少，或两个数异或值。

2-set

有 n 个互不相同的数，你需要将这 n 个数分成两个集合 A 和 B ，使得：

- 如果 x 在 A 的话，那么 $a - x$ 也要在 A 。
- 如果 x 在 B 的话，那么 $b - x$ 也要在 B 。

相同区间

有 n 个序列 $a[1 \dots M]$ ，其中 $0 \leq a[i] \leq K$ ，并且告诉你 m 条信息： $a[l1 \dots r1] = a[l2 \dots r2]$ ，求有几种满足条件的序列。