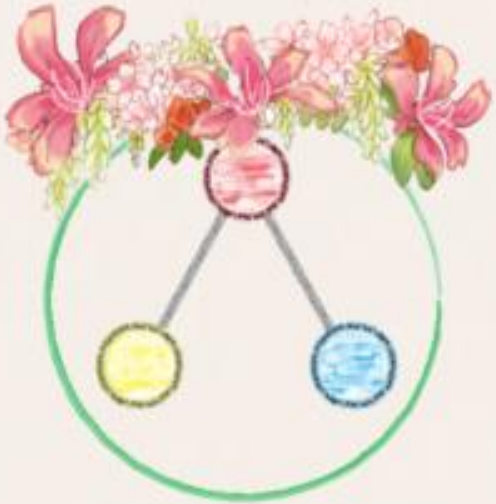


简单的连续段数据结构

A Simple Data Structure for Consecutive Sub-segments

WC2019营员交流

CommonAnts

陕西省 西北工业大学附属中学 刘承奥

作者水平十分有限，课件中难免错漏及不合理之处。
恳请各位批评指正。

引入

夜深了，我坐在床上。

可还不是能睡觉的时候，集训队作业的题还没有A，倘若不慎闭眼，本就渺茫的希望会更加淡的。

耷拉的手指随意点开一题，只见：

题目描述

给定一个整数1~n的排列A，A中的一个**区间**定义为：一个包含连续数字的连续子序列。更具体地，对于一对下标$a,b(1 \le a \le b \le n)$，子序列$A[a,b]=(A[a],A[a+1],\dots,A[b])$被称为一个**区间**，当且仅当，将其排序后，成为一个连续整数序列。例如，在排列$A=(3,1,7,5,6,4,2)$中，子序列$A[3,6]$是一个**区间**（它包含了整数4~7），而$A[1,3]$不是一个**区间**。

唉，我实在不想再求一遍强连通，或写一棵线段树了。

一定有什么办法的，授之以渔的办法……

引言和摘要

对于一个排列，我们称一个值域连续的区间为一个连续段。本文讨论它的性质与结构。并给出了线性时间内求出其树形结构的算法。

段，或者说区间，是序列上典型的组合结构。于是其在变换下的性质和结构也成了自然的话题。

这即是本文的讨论内容。具体而言，我们讨论段在某个置换下的不动点的性质、分布、结构、表示及相关算法。这些不动点可以由一棵树表示——我们称之为析合树。在通常的计算模型（RAM）下，求出给定置换的析合树仅需$O(n)$时间。

析合树可以用于解决许多问题，而其中来自信息学竞赛的并不鲜见。

定义

定义 1 <u>序列</u>：n个元素的有序集合，$n \in \mathbb{N}$，序列A的元素记作$A_i, i = 1, \ldots, n$，其中$A_i \le A_j$当且仅当$i \le j$；称n为<u>长度</u>或<u>阶</u>，i为A_i的<u>下标</u>。

对于n阶序列A：

定义 2 <u>段</u>（<u>区间</u>）：一个子集$S \subseteq A$，满足$(\nexists x, y, z)(A_x \le A_y \le A_z \wedge A_x \in S \wedge A_z \in S \wedge A_y \notin S)$；当$S$非空时分别称$S$的上下确界（和它们的下标）为其右端点$r$、左端点$l$，记作$S = [l, r]$；否则规定左端点无穷大、右端点无穷小。用$I$表示所有段的集合。

定义 3 <u>置换</u>（<u>排列</u>）：一个一一映射$p: A \to A, A_i \mapsto A_{p_i}$，也表示对应的$n$阶排列。

对于置换p：

定义 4 <u>不动点</u>（<u>连续段</u>）：一个段Q满足$Q \in I \wedge p[Q] \in I$。用$I_p$表示所有连续段的集合。一个段的值域$\mathrm{ran}(S) = [\min_{i \in S} p_i, \max_{i \in S} p_i]$（注意，映射$p$的值域用$p[S]$表示，与此不同），类似地$\mathrm{ran}^{-1}(S) = [\min_{i \in S} p_i^{-1}, \max_{i \in S} p_i^{-1}]$；

问题

对于n阶的A和p，I_p是什么样？

先把例子放上来：（忽略空集）

1表示$S \in I_p$，0则不然。

	[1,1]	[2,2]	[3,3]	[1,2]	[2,3]	[1,3]
1,2,3	1	1	1	1	1	1
1,3,2	1	1	1	0	1	1
2,1,3	1	1	1	1	0	1
2,3,1	1	1	1	1	0	1
3,1,2	1	1	1	0	1	1
3,2,1	1	1	1	1	1	1

性质

显然有许多平凡性质，如：

$$[i,i] \in I_p, [1,n] \in I_p$$

	[1,1]	[2,2]	[3,3]	[1,2]	[2,3]	[1,3]
1,2,3	1	1	1	1	1	1
1,3,2	1	1	1	0	1	1
2,1,3	1	1	1	1	0	1
2,3,1	1	1	1	1	0	1
3,1,2	1	1	1	0	1	1
3,2,1	1	1	1	1	1	1

还有：

引理 1 两个连续段的交是连续段。

$A \in I_p \wedge B \in I_p \rightarrow A \cap B \in I_p$

引理 2 两个相交连续段的并是连续段。

$A \in I_p \wedge B \in I_p \wedge A \cap B \neq \emptyset \rightarrow A \cup B \in I_p$

依据这两个引理，有些连续段可以被其它连续段表出，没有必要保留它们。

定义 5 如果连续段K能由连续段集X中的元素通过有限次交或相交并得出，称X（能）生成K；

定义 6 基：若连续段集合B能生成I_p中任意一个元素，称B是S的一个<u>**生成集**</u>，或者一个<u>**基**</u>。称一个基是<u>**极小**</u>的当且仅当它的真子集中不存在基。

基

引理 1 两个连续段的交是连续段。

引理 2 两个相交连续段的并是连续段。

如果连续段a能被连续段集合X中的元素通过有限次交或相交并得出，称X生成a；若连续段集合B能生成I_p中任意一个元素，称B是S的一个**<u>生成集</u>**，或者一个**<u>基</u>**。称一个基是**<u>极小</u>**的当且仅当它的真子集中不存在基。

如果两个连续段X,Y相交且并不互相包含，则它们可以生成$\{X\cap Y, X\cup Y\}$这两个互相包含的区间。能否只考虑$\{X\cap Y, X\cup Y\}$?

定义 7 <u>本原连续段</u>：对于一个连续段$X\in I_p$，如果不存在与之相交且互不包含的连续段（$(\exists Y)(Y\in I_p \wedge X\not\subseteq Y \wedge Y\not\subseteq X \wedge X\cap Y\neq \emptyset)$），我们就称$X$是**<u>本原连续段</u>**。所有本原连续段的集合称为M_p。

本原连续段与生成树

引理 3 对于n, A, p，存在一棵有根有序树$T = (V, E)$使得存在某个双射$f: V \to M_p$满足，$\forall u \in V$，$f(u) = \bigcup_{i \in son(u)} f(i)$；其中“∪”表示“连接”，即$[a, b] \cup [b+1, c] = [a, c]$，其余情况无意义；$son(u)$表示$u$所有孩子的有序集。

定义 8 称引理 3 中的树T为p的**生成树**。

引理 4 $\forall u \in V_T, son(u)$至少满足以下二者之一：

1. $son(u)$所有非平凡段的连接都是连续段；（$C_{4.1}$）
2. $son(u)$所有非平凡段的连接都不是连续段。（$C_{4.2}$）

其中非平凡意为长度不是0、1或$|son(u)|$。

~~即得易见平凡，仿照上例显然，留作习题答案略，读者自证不难。~~

~~反之亦然同理，推论自然成立，略去过程Q. E. D.，由上可知证毕。~~

如果每个位置都是一个非平凡连续段的右端点会怎么样？如果或有或无呢？

析合树

定义 9 如果本原连续段X满足$C_{4.1}\left(son\left(f^{-1}(X)\right)\right)$，称$X$是**正本原连续段**，否则称$X$是**异本原连续段**。

定义 10 如果点u满足$C_{4.1}(son(u))$，称u为**正点**（**合点**），否则称u为**异点**（**析点**）；两种点分别组成集合V_N和V_R。

我们还可以得到：

引理 5 生成树的叶节点个数为n；叶节点必然是合点；对于非叶节点，**一个合**点至少有2个孩子，一个析点至少有4个孩子。

定义 11 **析合树**：称满足引理 5 的有根有序树$T = ((V_N, V_R), E)$为n阶析合树；$R: I_p \mapsto T$表示连续段集与其对应的析合树的映射。

定理 1 R是双射。

析合树

引理 5 生成树的叶节点个数为n；叶节点必然是合点；对于非叶节点，**一个合点至少有2个孩子，一个析点至少有4个孩子。**

定义 11 析合树：称满足引理 5 的有根有序树$T = ((V_N, V_R), E)$为n阶析合树；$r: I_p \mapsto T$表示连续段集与其对应的析合树的映射。

我们现在知道：

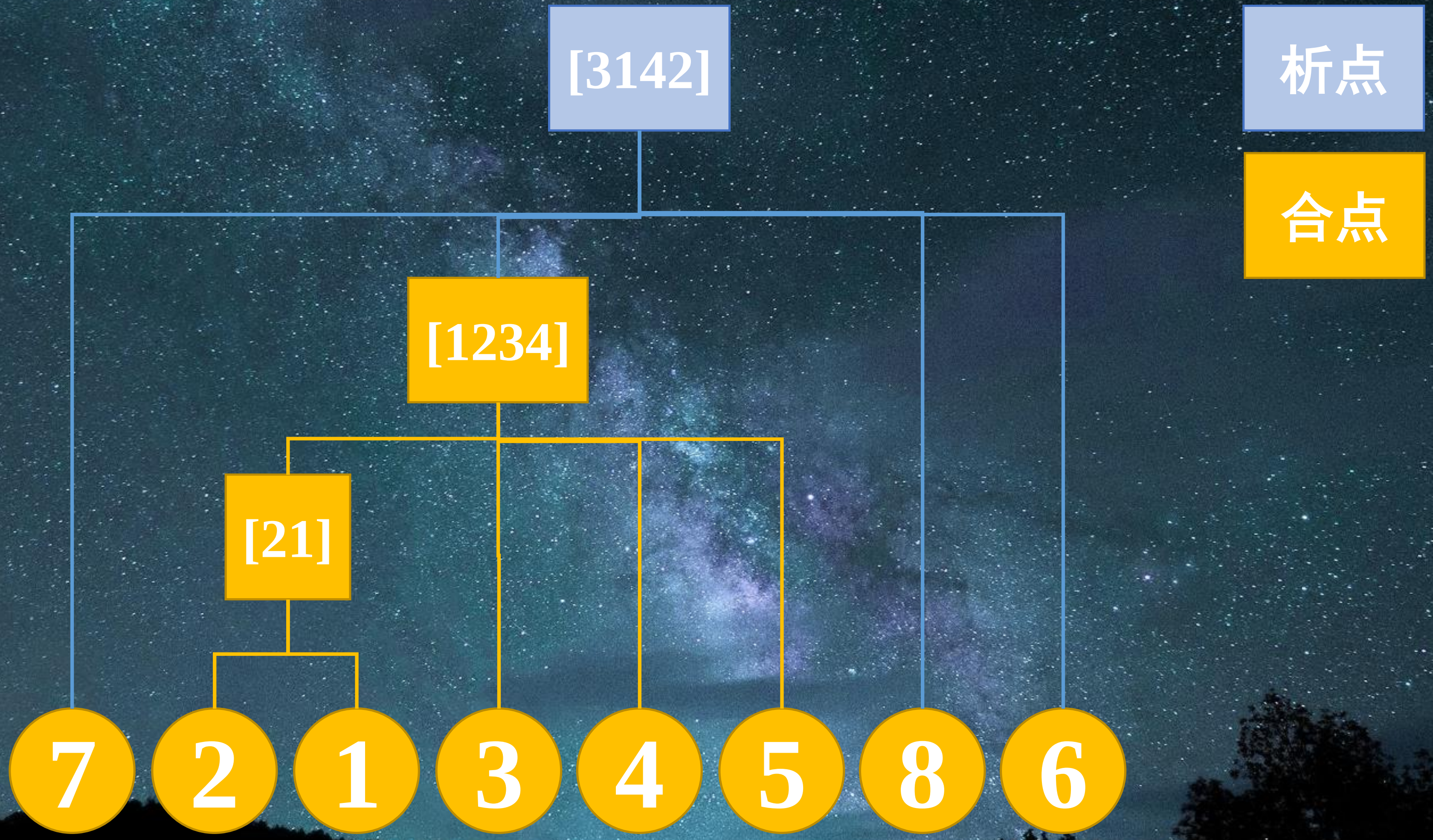
连续段集合可以等价地由一棵具有两种节点的有根有序树表示。

树要比连续段直观和经典得多。

稍微冷静一下，来看看它（几）的（道）应（水）用（题）：

析合树

例子：7 2 1 3 4 5 8 6



CERC2017 I. Intrinsic Interval.

引理 6 I_p是关于子集偏序的确界的格。

于是就有了此题和下一题。

题意：

给定置换p，多次（离线或在线）询问一个段，求I_p中最小包含它的段。

在$r(I_p)$上基本是个LCA问题。

修改版

题意：

给定置换p，多次（离线或在线）询问一个段，求I_p中最大包含于它的段。

在$r(I_p)$上有点复杂？

来源：袁方舟

ECFinal2018 B. Mysterious … Host.

题意：

给定n问I_p的种类数。

简单的树计数问题。

~~相信各位选手在近几年trivial计数的狂轰滥炸下早就会了。~~

~~再不济还能Min25BM不是，代数函数可都是p-recursive的。~~

CTSC2018 Day1 T3. 青蕈领主.

题意：

给定以每个元素为右端点的极大连续段的长度，求合法p个数。

尝试一下直接套……

然后需要计数……

~~Simple permutation，请~~

和原题差不多，大概只是省了分析结构的过程吧。这部分很平凡。

这个算法的一大缺陷，大约是Simple Permutation需要复合逆而尚未能快速求得。所以在$p \mapsto I_p$的部分常常束手无策……

构造算法

我们需要基于$r(I_p)$的数据结构。

给定p，如何求出$r(I_p)$？

这些题原本的解法？

构造算法

- $O(n\log n)$

似乎各种算法均如此。

试试点到区间连边的方法？

- 分块，分块，递归分块……
- $O(n\log\log n), O(n\log^* n), \dots, O(n\alpha n)$
- 一个经典的极限。

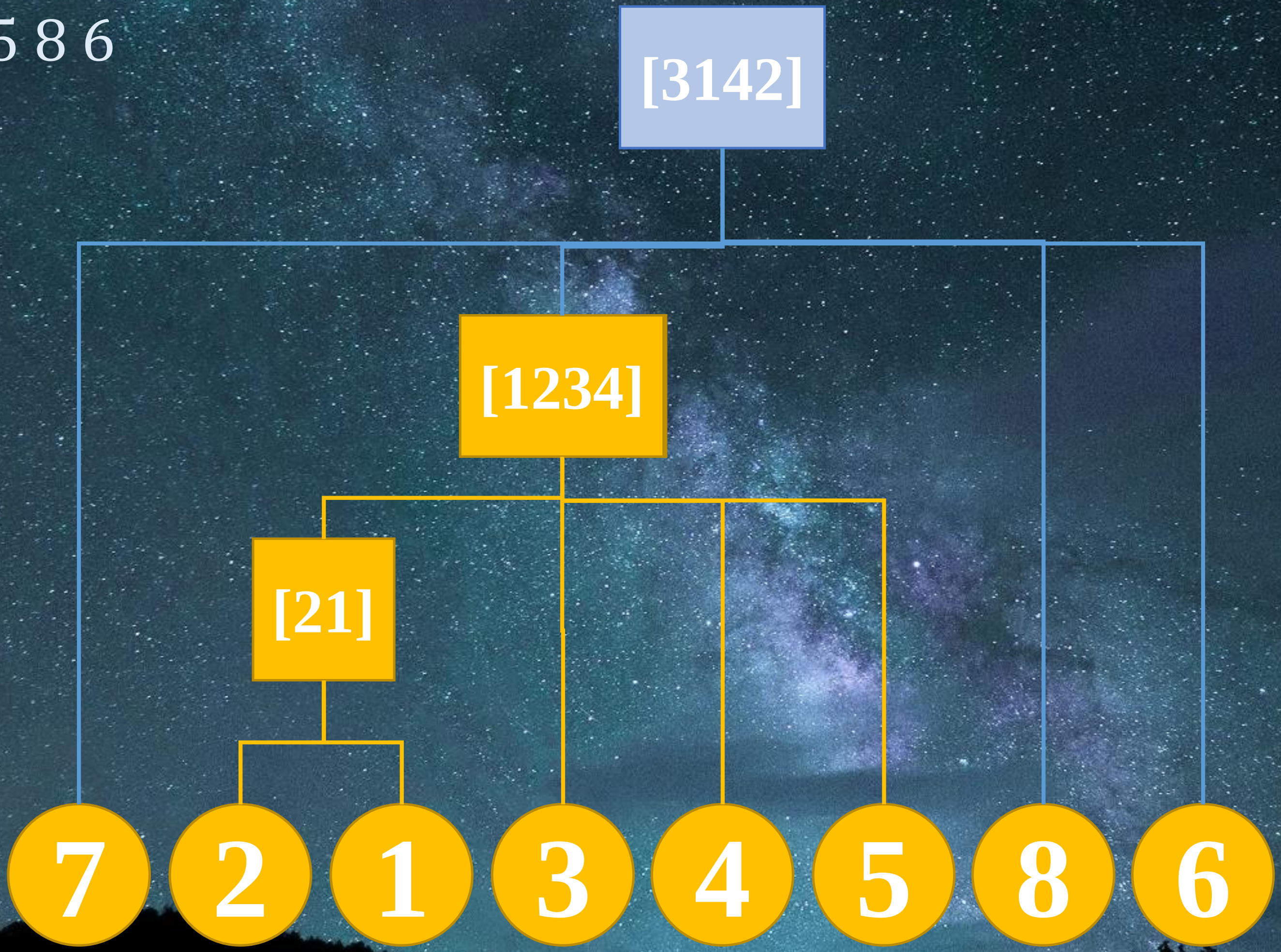
到尽头了吗？

置换并不是一个映射表。

朴素算法

从左向右插入点，维护当前的树。

例子：7 2 1 3 4 5 8 6



朴素算法

从左向右插入点，维护当前的树森林。

由于段是封闭的，**已有父亲**的点子树不会变化。析点子树不会变化。

新点有两种可能：

1. 成为合点的最后一个孩子。
2. 和最后若干个点合并产生一个新点。然后这个新点重复加入过程。

每次操作影响最后若干个点。

从后往前遍历，维护最值以判定连续段。

生成点时解决的子树个数等于扫描时间。

优秀的$O(n)$算法。

朴素算法

~~优秀的$O(n)$算法。~~

我差点信了。

生成点时解决的子树个数等于扫描时间。

如果生成不了呢？

例如只有一个析点的情况：1 5 2 6 3 7 4 8

- 每个点都要追溯到最开始。
- $O(n^2)$

改进

如何判断能否继续生成新的子树？

求出以每个元素为右端点的段中的极小左端点即可直接判断。

~~这不是《青蘁领主》的L[]吗？~~

然而找不到简便的线性算法。

再看看问题……

1. 我们没有利用算法中已经求得的信息（前缀的析合森林）
2. 也许可以退而求其次，求一个比L[]宽松的界，多判断几个子树，只要总数是$O(n)$的就行了。

思路

在前面的几个问题中有一种思路。

“给定置换p，多次（离线或在线）询问一个段，求I_p中最小包含它的段。”

由于$\mathrm{ran}([l,r]) = \bigcup_{i=l}^{r-1} \mathrm{ran}([i,i+1])$，可以从对应的$i$向$\mathrm{ran}^{-1}(\mathrm{ran}([i,i+1]))$连边并求传递闭包。

按引理6，假如我们现在插入的点是r，遍历到的左端点是l，那么以后合并的子树必然包含$\sup([l,r]) = [x,y]$。

$y > r$则不会再合并，否则下一个合并位置即为$\sup([l,r])$或包含其的合点。

~~好像没有线性算法啊！~~那是点到区间的传递性……

在原题中，求一个点能到达的范围需要$\sup([l,r])$的x,y两个端点。

但现在**必然有**$\boldsymbol{y = r}$。

改进

由于右端点皆为r，我们只需考虑左端点的传递性。

“左端点”即$\min\left(\operatorname{ran}^{-1}(\{p_i, p_{i+1}\})\right) = \min(p^{-1}_{p_i}, p^{-1}_{p_i+1}, \ldots, p^{-1}_{p_{i+1}})$这个RMQ。

容易线性解决（因为是离线问题，可以用Tarjan LCA之类的简便实现；但是我没有找到不用RMQ的简便方法）。右端点类似。

现在问题转化为，每个点i能够向左到达l_i以及路上的各点，问一个点直接和间接地向左最远能到达的位置L_i。我们称$[L_i, i]$为**左传递闭包**。

注意到$[L_i, i]$要么包含，要么相交（否则靠后者求并增大）。

其中极大的段组成了$[1, n]$的一个划分，其余段是它们的所有前缀。

从左到右遍历，用栈可以轻松求出。

新算法

上面算了这么多有什么用呢？

朴素算法中每个点r向左遍历到$[l,r]$时多了两条限制（剪枝）：

1. 如果存在某个 $i \in [l,r)$满足$r+1 \in \sup([i,i+1])$，接下来不可能存在连续段，直接结束遍历；
2. 否则，下一个连续段是$[\min_{i\in[l,r]} L_i, r]$。（$L_i$见上页）

真正的$O(n)$算法！

~~可以出线性题卡人了！~~

算法流程

1. 求出每个$\sup([i, i+1])$（用RMQ，转为笛卡尔树并用离线LCA）；
2. 求出每个点的左传递闭包（从左往右遍历，用栈）；
3. 用$\sup([i, i+1])$的右端点估计出以每个点为右端点的段的左端点的界；
4. 从左往右扫描，维护析合森林。其中每棵树引用前一棵树以及树中左传递闭包能到达的下一棵树。

修改

数据结构要维护数据，要增删查改。

能够支持哪些修改？

~~增删改~~

能支持保证$p_i \in [1, n]$且为互不相等的整数的增删吗？

先考虑插入。

修改

考虑每次插入导致析合树的改变，是一条链加上最多一个合点的孩子（作为叶子）。

左右链交错合并。

自闭。

有没有不交错的情况？

交换两个相邻p_i，左边的点修改右链，右边的点修改左链。

- ~~优秀的$\tilde{O}(n)$算法。~~

希望各位数据结构大佬能够解答我的疑惑。非常感谢！

更多问题及扩展

- 所以到底能怎么修改？
- 将子段换成其余组合结构，如树的连通子树，结果会有何变化？

- Simple Permutation？（不存在非平凡连续段的排列）

感谢

- 感谢CCF及广州二中提供营员交流的机会。
- 感谢张瑞喆教练、陈许旻学长、本校刘利丽老师和张明远、赵义凯、阮思凯、黄锦松等同学对我的帮助、培养和教导。
- 感谢王修涵、杨骏昭同学等在内容方面的讨论。
- 感谢艾莉芬特为本文验稿。
- 以及除此之外的人们。

参考资料

参考：

[1] 杨景钦, 《青蕈领主》命题报告, CTSC2018.
[2] 唐梓天, Interval解题报告, 2018第一轮集训队作业.
[3] 刘承奥, Mysterious … Host 题解, ECFinal2018.

题目：

[1] CTSC2018 Day1 T3. 青蕈领主.
[2] CERC2017 I. Intrinsic Interval.
[3] ECFinal2018 B. Mysterious … Host.
[4] CF997/493-1 E. Good Subsegments

Q&A

GL&HF

CommonAnts

陕西省 西北工业大学附属中学 刘承奥