

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

June 2016

大暴力经典解法

- 线段树平衡树
- 分块均衡
- 莫队
- 树(套树)*

fake Menu

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

大暴力经典解法

- 线段树平衡树
- 分块均衡
- 莫队
- 树(套树)*

奇技淫巧非经典解法

- 对时间分治及其拓展
- 二进制分组
- 整体二分

按时间分治

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

考虑一个常见的数据结构题模型：

按时间分治

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

考虑一个常见的数据结构题模型：

请实现一个数据结构支持以下操作：

- 询问（作出一个回答）
- 修改（对询问操作的答案造成影响）

按时间分治

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

考虑一个常见的数据结构题模型：

请实现一个数据结构支持以下操作：

- 询问（作出一个回答）
- 修改（对询问操作的答案造成影响）

对于满足以下条件的问题：

- 每个修改操作对询问的贡献相互独立；
- 允许离线。

我们可以用按时间分治进行处理。

按时间分治

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

如何利用修改独立，允许离线的性质？

按时间分治

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

如何利用修改独立，允许离线的性质？

设要处理的操作序列为 S ，把 S 等分成前后两部分 S_0 和 S_1 。

按时间分治

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分組

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

如何利用修改独立，允许离线的性质？

设要处理的操作序列为 S ，把 S 等分成前后两部分 S_0 和 S_1 。

显然的性质

- S_1 中的修改对 S_0 中的询问不产生影响。
- S_1 中的询问受 S_0 中所有的修改 和 S_1 中在它之前的修改 的影响。

按时间分治

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

如何利用修改独立，允许离线的性质？

设要处理的操作序列为 S ，把 S 等分成前后两部分 S_0 和 S_1 。

显然的性质

- S_1 中的修改对 S_0 中的询问不产生影响。
- S_1 中的询问受 S_0 中所有的修改 和 S_1 中在它之前的修改 的影响。

因此我们可以递归处理 S_0 和 S_1 ，然后再处理 S_0 中的修改对 S_1 中的询问的影响即可。

按时间分治

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

给一个伪代码：

```
def Divide_and_Conquer(S):  
    if S.length() == 1:  
        return  
    Divide S into [S0, S1]  
    Divide_and_Conquer(S0)  
    Divide_and_Conquer(S1)  
    Update queries in S1 with modifies in S0
```

按时间分治

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

我们看到，按时间分治的关键问题在于如何处理
Update queries in S_1 with modifies in S_0

按时间分治

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

我们看到，按时间分治的关键问题在于如何处理

Update queries in S_1 with modifies in S_0

这种情况下，询问在所有的修改之后，原问题的动态修改操作不复存在。

转化成“一开始给出所有修改，然后要求回答若干询问”的问题。

按时间分治

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

下面分析这个做法的时间复杂度：

按时间分治

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

下面分析这个做法的时间复杂度：
假设解决“无动态修改操作的原问题”的复杂度为 $O(f(n))$

按时间分治

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

下面分析这个做法的时间复杂度：

假设解决“无动态修改操作的原问题”的复杂度为 $O(f(n))$
那么按时间分治的复杂度为

$$T(n) = 2T(\frac{n}{2}) + O(f(n))$$

由主定理得

$$T(n) = O(f(n) \log n)$$

按时间分治

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心理念

复杂度分析

简单例题

拓展

二进制分组

核心理念

复杂度分析

简单例题

整体二分

核心理念

复杂度分析

简单例题

总结

Extra

下面分析这个做法的时间复杂度：

假设解决“无动态修改操作的原问题”的复杂度为 $O(f(n))$
那么按时间分治的复杂度为

$$T(n) = 2T(\frac{n}{2}) + O(f(n))$$

由主定理得

$$T(n) = O(f(n) \log n)$$

因此我们用 $\log n$ 的代价把原问题的动态操作去掉了。

按时间分治

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心理念

复杂度分析

简单例题

拓展

二进制分组

核心理念

复杂度分析

简单例题

整体二分

核心理念

复杂度分析

简单例题

总结

Extra

下面分析这个做法的时间复杂度：

假设解决“无动态修改操作的原问题”的复杂度为 $O(f(n))$
那么按时间分治的复杂度为

$$T(n) = 2T(\frac{n}{2}) + O(f(n))$$

由主定理得

$$T(n) = O(f(n) \log n)$$

因此我们用 $\log n$ 的代价把原问题的动态操作去掉了。

注意这里要求单次处理的复杂度只与操作个数有关，否则复杂度会退化。

共点圆

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

BZOJ2961

题意简化以后长这样：

维护一个平面，支持：

- 插入一个点
- 给一个半平面，询问是否所有点都在这个半平面内

允许离线。

共点圆

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

BZOJ2961

题意简化以后长这样：

维护一个平面，支持：

- 插入一个点
- 给一个半平面，询问是否所有点都在这个半平面内

允许离线。

满足“修改独立，允许离线”的性质。

共点圆

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

我们直接套用按时间分治，问题简化为：

- 开始时给定一些点
- 每次给一个半平面，询问是否所有点都在这个半平面内

共点圆

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

我们直接套用按时间分治，问题简化为：

- 开始时给定一些点
- 每次给一个半平面，询问是否所有点都在这个半平面内

我们可以求出所有点的凸包，再把所有半平面按斜率排序然后在凸包上扫一遍就可以了。

共点圆

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心理想

复杂度分析

简单例题

拓展

二进制分治

核心理想

复杂度分析

简单例题

整体二分

核心理想

复杂度分析

简单例题

总结

Extra

我们直接套用按时间分治，问题简化为：

- 开始时给定一些点
- 每次给一个半平面，询问是否所有点都在这个半平面内

我们可以求出所有点的凸包，再把所有半平面按斜率排序然后在凸包上扫一遍就可以了。

复杂度 $O(n \log n)$ ，套上按时间分治后为 $O(n \log^2 n)$

共点圆

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心理想

复杂度分析

简单例题

拓展

二进制分组

核心理想

复杂度分析

简单例题

整体二分

核心理想

复杂度分析

简单例题

总结

Extra

TLE 了?

共点圆

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

TLE 了?
利用分治的性质!

共点圆

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

TLE 了？

利用分治的性质！

注意到斜率排序可以归并，求凸包需要的排序也可以归并。

因此复杂度降为 $O(n \log n)$

共点圆

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

TLE 了?

利用分治的性质!

注意到斜率排序可以归并, 求凸包需要的排序也可以归并。

因此复杂度降为 $O(n \log n)$

DMD's Code: <http://paste.ubuntu.com/17795583/>

动态半平面交

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分治

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

维护一个平面，支持：

- 插入一个半平面
- 删除一个半平面
- 查询一个点是否在当前的半平面交内

动态半平面交

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分組

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

维护一个平面，支持：

- 插入一个半平面
- 删除一个半平面
- 查询一个点是否在当前的半平面交内

存在删除操作，修改不独立了！

动态半平面交

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分組

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

维护一个平面，支持：

- 插入一个半平面
- 删除一个半平面
- 查询一个点是否在当前的半平面交内

存在删除操作，修改不独立了！
然而我们仍然使用按时间分治解决。

动态半平面交

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心理想

复杂度分析

简单例题

拓展

二进制分组

核心理想

复杂度分析

简单例题

整体二分

核心理想

复杂度分析

简单例题

总结

Extra

将操作序列 S 等分为 S_0 和 S_1 。

动态半平面交

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

将操作序列 S 等分为 S_0 和 S_1 。

对于 S_0 中的查询，它们的与 S_1 中的操作完全无关，可以递归解决。

动态半平面交

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

将操作序列 S 等分为 S_0 和 S_1 。

对于 S_0 中的查询，它们的与 S_1 中的操作完全无关，可以递归解决。

对于 S_1 中的查询，它们的答案与两部分操作有关：

- S_0 中未被删除的部分
- S_1 中在此查询之前未被删除的部分

动态半平面交

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心理想

复杂度分析

简单例题

拓展

二进制分组

核心理想

复杂度分析

简单例题

整体二分

核心理想

复杂度分析

简单例题

总结

Extra

将操作序列 S 等分为 S_0 和 S_1 。

对于 S_0 中的查询，它们的与 S_1 中的操作完全无关，可以递归解决。

对于 S_1 中的查询，它们的答案与两部分操作有关：

- S_0 中未被删除的部分
- S_1 中在此查询之前未被删除的部分

对于第二个部分可以递归解决。

实际要处理的部分是 “ S_0 中未被删除的部分” 对 “ S_1 中的询问” 的贡献。

动态半平面交

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

转化为：

开始给定一组半平面，要求支持：

- 删除一个半平面
- 查询一个点是否在当前的半平面交内

动态半平面交

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

转化为：

开始给定一组半平面，要求支持：

- 删除一个半平面
- 查询一个点是否在当前的半平面交内

只有删除操作？时间倒流！

动态半平面交

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

转化为：

开始给定一组半平面，要求支持：

- 删除一个半平面
- 查询一个点是否在当前的半平面交内

只有删除操作？时间倒流！

我们可以先求出最后没有被删除的半平面有哪些，然后逆序处理每一个操作。

查询没有变，删除变成了插入。

动态半平面交

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

转化后：

维护一个平面，要求支持：

- 插入一个半平面
- 查询一个点是否在当前的半平面交内

动态半平面交

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

转化后：

维护一个平面，要求支持：

- 插入一个半平面
- 查询一个点是否在当前的半平面交内

再次对时间分治！

动态半平面交

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心理想

复杂度分析

简单例题

拓展

二进制分组

核心理想

复杂度分析

简单例题

整体二分

核心理想

复杂度分析

简单例题

总结

Extra

转化后：

维护一个平面，要求支持：

- 插入一个半平面
- 查询一个点是否在当前的半平面交内

再次对时间分治！

每次求出前半部分的半平面交，对后半部分的询问进行判定。

可以在 $O(n \log n)$ 内实现。 How?

动态半平面交

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心理想

复杂度分析

简单例题

拓展

二进制分组

核心理想

复杂度分析

简单例题

整体二分

核心理想

复杂度分析

简单例题

总结

Extra

转化后：

维护一个平面，要求支持：

- 插入一个半平面
- 查询一个点是否在当前的半平面交内

再次对时间分治！

每次求出前半部分的半平面交，对后半部分的询问进行判定。

可以在 $O(n \log n)$ 内实现。 How?

问题解决，总复杂度 $O(n \log^2 n)$ 。

动态半平面交

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心理想

复杂度分析

简单例题

拓展

二进制分组

核心理想

复杂度分析

简单例题

整体二分

核心理想

复杂度分析

简单例题

总结

Extra

转化后：

维护一个平面，要求支持：

- 插入一个半平面
- 查询一个点是否在当前的半平面交内

再次对时间分治！

每次求出前半部分的半平面交，对后半部分的询问进行判定。

可以在 $O(n \log n)$ 内实现。 How?

问题解决，总复杂度 $O(n \log^2 n)$ 。

论外：强制在线怎么做？

动态半平面交

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心理想

复杂度分析

简单例题

拓展

二进制分组

核心理想

复杂度分析

简单例题

整体二分

核心理想

复杂度分析

简单例题

总结

Extra

转化后：

维护一个平面，要求支持：

- 插入一个半平面
- 查询一个点是否在当前的半平面交内

再次对时间分治！

每次求出前半部分的半平面交，对后半部分的询问进行判定。

可以在 $O(n \log n)$ 内实现。 How?

问题解决，总复杂度 $O(n \log^2 n)$ 。

论外：强制在线怎么做？

线段树套可持久化平衡树

二进制分组

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

“修改独立，要求在线”的问题，可以用二进制分组解决。

二进制分组

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

“修改独立，要求在线”的问题，可以用二进制分组解决。
考虑树状数组的数据组织形式：
 BIT_i 存储区间 $[i - lowbit(i) + 1, i]$ 的信息。

二进制分组

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

“修改独立，要求在线”的问题，可以用二进制分组解决。

考虑树状数组的数据组织形式：

BIT_i 存储区间 $[i - lowbit(i) + 1, i]$ 的信息。

模仿树状数组的分组方式：

在第 i 个修改时，把区间 $[i - lowbit(i) + 1, i]$ 内的修改信息分为一组，用数据结构维护。

询问时在划分成的 $O(\log n)$ 个组内分别询问。

二进制分组

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

设用数据结构维护长度为 n 的修改序列 $O(f(n))$ ，单次询问 $O(g(n))$

二进制分组

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心理想

复杂度分析

简单例题

拓展

二进制分组

核心理想

复杂度分析

简单例题

整体二分

核心理想

复杂度分析

简单例题

总结

Extra

设用数据结构维护长度为 n 的修改序列 $O(f(n))$ ，单次询问 $O(g(n))$
第 i 次修改的复杂度是 $O(f(\text{lowbit}(i)))$

二进制分组

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心理想

复杂度分析

简单例题

拓展

二进制分组

核心理想

复杂度分析

简单例题

整体二分

核心理想

复杂度分析

简单例题

总结

Extra

设用数据结构维护长度为 n 的修改序列 $O(f(n))$ ，单次询问 $O(g(n))$

第 i 次修改的复杂度是 $O(f(\text{lowbit}(i)))$

所有修改的总复杂度是

$$\sum_{i=1}^n O(f(\text{lowbit}(i))) \leq O(f(n) \log n) \quad \underline{\text{*Why?}}$$

二进制分组

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

设用数据结构维护长度为 n 的修改序列 $O(f(n))$ ，单次询问 $O(g(n))$

第 i 次修改的复杂度是 $O(f(\text{lowbit}(i)))$

所有修改的总复杂度是

$$\sum_{i=1}^n O(f(\text{lowbit}(i))) \leq O(f(n) \log n) \quad \underline{\text{*Why?}}$$

单次查询的复杂度是 $O(g(n) \log n)$

二进制分组

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

设用数据结构维护长度为 n 的修改序列 $O(f(n))$ ，单次询问 $O(g(n))$

第 i 次修改的复杂度是 $O(f(\text{lowbit}(i)))$

所有修改的总复杂度是

$$\sum_{i=1}^n O(f(\text{lowbit}(i))) \leq O(f(n) \log n) \quad \underline{\text{*Why?}}$$

单次查询的复杂度是 $O(g(n) \log n)$

因此，使用二进制分组，我们以 $\log n$ 的代价把原问题的动态修改去掉了，并且可以支持在线。

共点圆加强版

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

BZOJ4140

维护一个平面，支持：

- 插入一个点
- 给一个半平面，询问是否所有点都在这个半平面内

强制在线。

共点圆加强版

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

BZOJ4140

维护一个平面，支持：

- 插入一个点
- 给一个半平面，询问是否所有点都在这个半平面内

强制在线。

直接二进制分组，每组内求凸包，查询需要在凸包上二分。
修改总复杂度 $O(n \log^2 n)$ ，单次查询 $O(\log^2 n)$ 。
代码奇短，常数巨小，跑的比 $O(n \log n)$ 的平衡树还快。

共点圆加强版

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

BZOJ4140

维护一个平面，支持：

- 插入一个点
- 给一个半平面，询问是否所有点都在这个半平面内

强制在线。

直接二进制分组，每组内求凸包，查询需要在凸包上二分。

修改总复杂度 $O(n \log^2 n)$ ，单次查询 $O(\log^2 n)$ 。

代码奇短，常数巨小，跑的比 $O(n \log n)$ 的平衡树还快。

其实建凸包可以少一个 \log *How?

但不影响总复杂度

共点圆加强版

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

按时间分治

二、进制分组

简单例题

整体二分

总结

BZOJ4140

维护一个平面，支持：

- 插入一个点
- 给一个半平面，询问是否所有点都在这个半平面内

强制在线。

直接二进制分组，每组内求凸包，查询需要在凸包上二分。

修改总复杂度 $O(n \log^2 n)$, 单次查询 $O(\log^2 n)$ 。

代码奇短，常数巨小，跑的比 $O(n \log n)$ 的平衡树还快。

其实建凸包可以少一个 \log *How?

但不影响总复杂度

%Clariss

<http://www.cnblogs.com/clrs97/p/5008300.html>

共点圆加强版 *

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

维护一个平面，支持：

- 插入一个点
- 给一个半平面，询问是否区间 $[l, r]$ 内所有点都在这个半平面内

强制在线。

共点圆加强版 *

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

维护一个平面，支持：

- 插入一个点
- 给一个半平面，询问是否区间 $[l, r]$ 内所有点都在这个半平面内

强制在线。

区间查询怎么玩？

共点圆加强版 *

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

维护一个平面，支持：

- 插入一个点
- 给一个半平面，询问是否区间 $[l, r]$ 内所有点都在这个半平面内

强制在线。

区间查询怎么玩？

改变分组方式，建成一棵线段树。

*How?

共点圆加强版 *

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

维护一个平面，支持：

- 插入一个点
- 给一个半平面，询问是否区间 $[l, r]$ 内所有点都在这个半平面内

强制在线。

区间查询怎么玩？

改变分组方式，建成一棵线段树。
时间复杂度不变，空间多一个 \log 。

*How?

*Why?

共点圆加强版 *

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

维护一个平面，支持：

- 插入一个点
- 给一个半平面，询问是否区间 $[l, r]$ 内所有点都在这个半平面内

强制在线。

区间查询怎么玩？

改变分组方式，建成一棵线段树。
时间复杂度不变，空间多一个 \log 。
来张图演示一下？

*How?

*Why?

整体二分

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

二分答案是常见的技巧，然而在数据结构题中往往不奏效，原因是我们往往需要预处理才能回答当前判定点的答案。而回答多组询问时，由于每个答案都有可能被二分到，预处理的复杂度就不可接受了。

整体二分

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分組

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

二分答案是常见的技巧，然而在数据结构题中往往不奏效，原因是我们往往需要预处理才能回答当前判定点的答案。而回答多组询问时，由于每个答案都有可能被二分到，预处理的复杂度就不可接受了。整体二分可以理解为“对多组询问同时二分”，可以用来解决这个问题。

整体二分

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分組

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

整体二分需要题目满足以下性质：

- 询问答案可二分
- 允许离线
- 修改对判定答案的贡献相互独立
- 修改对判定答案的贡献与当前二分值无关，具有可加性

整体二分

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心理想

复杂度分析

简单例题

拓展

二进制分組

核心理想

复杂度分析

简单例题

整体二分

核心理想

复杂度分析

简单例题

总结

Extra

整体二分需要题目满足以下性质：

- 询问答案可二分
- 允许离线
- 修改对判定答案的贡献相互独立
- 修改对判定答案的贡献与当前二分值无关，具有可加性

注意到由于最后一条性质，如果我们计算出某些修改对询问的贡献，那么这个贡献永远不会改变。

当二分值改变时，我们也不需要再次计算这部分修改的贡献。这保证了整体二分的复杂度。

整体二分

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分組

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

因为可能不好理解，所以先把伪代码贴上来

```
def Divide_and_Conquer(Q, L, R):  
    # Q: queries to handle  
    # [L, R]: current range of answer  
    if L == R: # terminate  
        for each q in Q:  
            q.ans = L  
        return  
    M = (L + R) / 2  
    update Q with modifications in range [L, M] && put results into .res  
    for each q in Q:  
        if q.goal < q.current + q.res: # answer is in [L, M]  
            Q1.append(q)  
        else: # answer is in [M + 1, R]  
            # to avoid duplicate work  
            q.current += q.res  
            Q2.append(q)  
    Divide_and_Conquer(Q1, L, M) # recursion  
    Divide_and_Conquer(Q2, M + 1, R)
```

复杂度分析

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

定义 $F(C, S)$ 表示二分区间长 C ，处理 S 个询问的复杂度，

复杂度分析

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分組

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

定义 $F(C, S)$ 表示二分区间长 C ，处理 S 个询问的复杂度，则有

$$T(C, S) = T\left(\frac{C}{2}, S_0\right) + T\left(\frac{C}{2}, S_1\right) + O(f(S))$$

解得 $T(C, n) \leq O(f(n) \log C)$

复杂度分析

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分組

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

定义 $F(C, S)$ 表示二分区间长 C ，处理 S 个询问的复杂度，则有

$$T(C, S) = T(\frac{C}{2}, S_0) + T(\frac{C}{2}, S_1) + O(f(S))$$

$$\text{解得 } T(C, n) \leq O(f(n) \log C)$$

注意这里要求单次处理询问的复杂度只与询问个数有关，否则复杂度会退化。

经典问题

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

给定一个序列，每次查询区间 k 小值。

经典问题

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心理想

复杂度分析

简单例题

拓展

二进制分組

核心理想

复杂度分析

简单例题

整体二分

核心理想

复杂度分析

简单例题

总结

Extra

给定一个序列，每次查询区间 k 小值。
直接上整体二分，用树状数组维护。
 $O((n + Q) \log n \log C)$

How?

经典问题 *

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

BZOJ2738 给定一个矩阵，每次查询子矩阵 k 小值。

经典问题 *

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

BZOJ2738 给定一个矩阵，每次查询子矩阵 k 小值。
这有什么区别吗？
 $O((n^2 + Q) \log^2 n \log C)$

经典问题 *

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

BZOJ2738 给定一个矩阵，每次查询子矩阵 k 小值。
这有什么区别吗？

$O((n^2 + Q) \log^2 n \log C)$

IzumiKonata's Code:

<http://paste.ubuntu.com/17795357/>

ZJOI2013 K 大数查询

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心理想

复杂度分析

简单例题

拓展

二进制分治

核心理想

复杂度分析

简单例题

整体二分

核心理想

复杂度分析

简单例题

总结

Extra

BZOJ3110

维护一个可重集序列，支持以下操作：

- 对区间 $[l, r]$ 内的每个可重集插入一个数 x
- 查询区间 $[l, r]$ 内的可重集的并集的第 k 大数

ZJOI2013 K 大数查询

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心理想

复杂度分析

简单例题

拓展

二进制分治

核心理想

复杂度分析

简单例题

整体二分

核心理想

复杂度分析

简单例题

总结

Extra

BZOJ3110

维护一个可重集序列，支持以下操作：

- 对区间 $[l, r]$ 内的每个可重集插入一个数 x
- 查询区间 $[l, r]$ 内的可重集的并集的第 k 大数

先上整体二分。

接下来我们要处理“某些询问对某些修改”的贡献。

ZJOI2013 K 大数查询

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

问题转化为

维护一个序列，支持以下操作：

- 区间 $[l, r]$ 每个数 $+1$
- 查询区间和

特殊要求：复杂度只与操作个数有关

ZJOI2013 K 大数查询

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

问题转化为

维护一个序列，支持以下操作：

- 区间 $[l, r]$ 每个数 $+1$
- 查询区间和

特殊要求：复杂度只与操作个数有关

线段树？ $O(q \log C)$

ZJOI2013 K 大数查询

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

问题转化为

维护一个序列，支持以下操作：

- 区间 $[l, r]$ 每个数 $+1$
- 查询区间和

特殊要求：复杂度只与操作个数有关

线段树？ $O(q \log C)$

注意到“修改独立，允许离线”，可以按时间分治。

然后排序乱搞就可以了。用归并 $O(q \log q)$

ZJOI2013 K 大数查询

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

问题转化为

维护一个序列，支持以下操作：

- 区间 $[l, r]$ 每个数 $+1$
- 查询区间和

特殊要求：复杂度只与操作个数有关

线段树？ $O(q \log C)$

注意到“修改独立，允许离线”，可以按时间分治。

然后排序乱搞就可以了。用归并 $O(q \log q)$

总复杂度 $O(n \log^2 n)$ ，常数比树套树小。

总结

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

按时间分治、二进制分组和整体二分都是解决数据结构题的有效工具。
其实质是利用了数据结构题中操作的特殊性质。
适当地运用这些算法，可以转化、简化问题，降低代码难度。
这启示我们，要充分挖掘和利用题目特殊性，从而得到更加优美的算法。

Extra: 动态半平面交 ++

浅析《浅谈数据结构题的几个非经典解法》

IzumiKonata

Menu

按时间分治

核心思想

复杂度分析

简单例题

拓展

二进制分组

核心思想

复杂度分析

简单例题

整体二分

核心思想

复杂度分析

简单例题

总结

Extra

维护一个平面，支持：

- 插入一个带权的半平面
- 删除一个半平面
- 给一个点，查询覆盖了此点的权值最大的半平面