

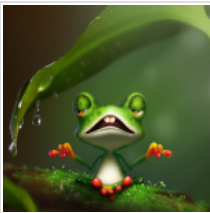
# SprintfWater的专栏

目录视图

摘要视图

RSS 订阅

## 个人资料



SprintfWater



访问: 227771次

积分: 3902

等级: BLOG &gt; 5

排名: 第3298名

原创: 133篇 转载: 196篇

译文: 3篇 评论: 51条

## 文章搜索

## 文章分类

- 数据结构 (13)
- 状态压缩 (6)
- 网络流 (8)
- 动态规划 (DP) (19)
- 二分匹配 (3)
- 树状数组 (1)
- 线段树 (4)
- 字符串处理 (13)
- 计算几何 (2)
- 图论搜索 (8)
- 数论 (3)
- 综合性水题 (3)
- C++编程 (43)
- STL标准模板库 (10)
- 设计模式 (2)
- Qt相关资料 (22)
- linux kernel (9)
- linux开发环境 (46)
- linux API编程 (31)
- 多核多cpu编程 (3)
- linux网络编程 (30)
- 数据库编程SQL (15)

[微信公众平台开发入门](#) [Markdown编辑器轻松写博文](#) [软考,你过了吗?](#) [天天爱答题 一大波C币向你袭来](#) [读文章说感想送好书](#)

## 个人对主席树算法的理解

分类: 数据结构

2013-06-24 15:17

3733人阅读

评论(2)

收藏

举报

首先借主席树发明人的一段话:

[html] view plain copy print ?

- ..这个东西是当初我弱不会划分树的时候写出来替代的一个玩意..被一小撮别有用心的取了很奇怪的名字 > <
- 想法是对原序列的每一个前缀[1..i]建立出一颗线段树维护值域上每个数的出现次数,然后发现这样的树是可以减的,然后就没有然后了

转载请注明出处,谢谢。

<http://blog.csdn.net/sprintfwater/article/details/9162041>

首先定义:

[cpp] view plain copy print ?

```
01. a[MAXN], a2[MAXN];
02. struct node
03. {
04.     node *ch[2];
05.     int siz;
06.     node(){ch[0]=ch[1]=NULL; siz=0;}
07.     node(node *_ch0, node *_ch1, int _siz):siz(_siz){ch[0]=_ch0, ch[1]=_ch1;}
08.     void update()
09.     {
10.         if (ch[0]) siz+=ch[0]->siz;
11.         if (ch[1]) siz+=ch[1]->siz;
12.     }
13. }*null=new node(), *root[MAXN]={NULL};
```

自己理解:

- 前提条件: 有N个数字, size个不同的数字。
- 用a[N]存储原来的数字, a2[size]存储排好序的数字;
- 也就是每一个点root[i]都维护着size大小的线段树, 而该线段树维护的信息为: 在a[1]到a[i]数字集合中, size种数字分别出现的次数。
- root[i] -> ch[0] 保存的是排好序, 前size/2种数字分别出现的次数, root[i] -> ch[1] 保存的是排好序, 后size/2种数字分别出现的次数;

值得注意的是, 并不是每个root[i]都必须重新开辟size \* log2(size)的空间, 比如当新加入的a[i]非常小, 那么root[i] -> ch[1]后size/2种数字分别出现的次数

相对于root[i-1] -> ch[1]是不会变。所以可以只用有孩子指针指向root[i-1] -> ch[1]所指的节点就可以共用。当新加入的a[i]非常大的时候, 那么ch[0]就可以共用。

建树的关键代码:

分布式系统（云存储&云计算） (30)

脚本语言(python) (3)

图形学 (13)

嵌入式技术积累 (1)

网站资料 (5)

Android应用开发 (5)

UML建模 (3)

智力游戏 (7)

dpgk (0)

路由配置 (0)

web服务 (1)

dns域名系统 (1)

文章存档

2015年03月 (1)

2015年02月 (1)

2015年01月 (8)

2014年11月 (4)

2014年10月 (1)

展开

阅读排行

2013腾讯面经（技术类-） (6045)

Linux命令——chmod(修 (5849)

C程序访问hadoop程序练 (3966)

个人对主席树算法的理解 (3730)

HDFS: c/c++接口 libhdfs (3427)

Ubuntu 下gvim/vim 安装 (3243)

我的网络流sap, isap, t (3021)

stl map用法和make\_pai (2955)

一个进程能用的最大内存 (2697)

OSI七层协议模型和TCP/IP (2659)

评论排行

hdu 4541 Ten Googol (5)

2013腾讯面经（技术类-） (4)

linux 终端操作mysql的一 (3)

hdu 4504威威猫系列故事 (3)

C程序访问hadoop程序练 (3)

C++ const函数返回值必 (2)

hdu 4512 吉哥系列故事- (2)

我的git 基本操作 (2)

个人对主席树算法的理解 (2)

android游戏开发须知! (2)

推荐文章

\* 【ShaderToy】开篇

\* FFmpeg源代码简单分析:

avio\_open2()

\* 技能树之旅: 从模块分离到测试

\* Qt5官方demo解析集36——Wiggly

Example

\* Unity3d HDR和Bloom效果（高动

态范围图像和泛光）

\* Android的Google官方设计指南

最新评论

[cpp] view plain copy print ?

```
01. void make_node(node *&y,node *&x,int l,int r,int t)
02. {
03.     if (x==NULL) x=null;
04.     y=new node();
05.     int m=(l+r)>>1;
06.     if (l==r)//已经到达叶子节点,
07.     {
08.         *y=*x;
09.         y->siz++;
10.         return;
11.     }
12.     if (t<=a2[m])
13.     {
14.         make_node(y->ch[0],x->ch[0],l,m,t);
15.         y->ch[1]=x->ch[1];
16.         y->update();
17.     }
18.     else
19.     {
20.         make_node(y->ch[1],x->ch[1],m+1,r,t);
21.         y->ch[0]=x->ch[0];
22.         y->update();
23.     }
24. }
```

为了防止new爆T，那可以先用数组存好。

一：先分析空间复杂度：

每加入一个a【i】时，就会增加log2（size）个新节点，也就是每一层会产生一个新节点。所以空间复杂度为：  
N \* log2(size)

二：当查看一个一个区间中的 第K大值时

在第一层时：

1.先查看这段区间在前size/2个的出现次数是否大于k，如果是，那么第k个，肯定在前size/2中，否则在后size/2中，

那么第k大，就得减去前size/2个出现的次数之和，变为在后size/2种数字中求第(k - 前size/2个出现的次数之和)大

2. 而前size/2个的出现次数可以直接相减可得，比如要看区间L到R，前size/2种数字出现的次数之和，  
等于(root[R] -> ch[0] -> size) - (root[L - 1] -> ch[0] -> size)

3.往下层走，只不过就是size变为了size/2，递归下去就可以了。

很容易看出查询一次的时间复杂度为log2(size);

关键代码如下：

[cpp] view plain copy print ?

```
01. void find(node *&x1,node *&x2,int l,int r,int k)
02. {
03.     if (x1==NULL) x1=null;
04.     if (x2==NULL) x2=null;
05.     if (l==r) {printf("%d\n",a2[l]);return;}
06.     int m=(l+r)>>1;
07.     int ls=l;
08.     if (x2->ch[0] !=>ch[0]->siz;
09.     if (x1->ch[0] !=>ch[0]->siz;
10.
11.     if (ls>=k) find(x1->ch[0],x2->ch[0],l,m,k);
12.     else find(x1->ch[1],x2->ch[1],m+1,r,k-ls);
13. }
```

下面是网上转载的主席树完整代码：

2013腾讯面经（技术类-后台开发 qq\_20497909: 能列个书单吗

QTextEdit捕捉回车信号 aa642796204: 顶，就是不知道怎样处理回车建了，希望大神来顶下。

个人对主席树算法的理解 自在\_飞花: “值得注意的是，并不是每个root【i】都必须重新开辟size \* log2(size)的空间，”...

2013腾讯面经（技术类-后台开发 zsd1345a780rfajwet: 同学，其实你的ACM学习经历比硕士学历有用多了，我是今年毕业的硕士，没有参加过这些竞赛，只是跟老师做...

2013腾讯面经（技术类-后台开发 爱情魔法师: 楼主 太牛了，好像请教一下你啊，我16年毕业，还有一年多，但是不知该学些什么才能进腾讯，能加个qq...

11988 - Broken Keyboard (a.k.a U\_\_F\_\_O: 请问这个程序可以去哪里检验下？因为PAT好像就考试的时候进行检验，我找不到可以做这题的地方。

后缀数组模板 \_KDH: 为什么r的末尾一定要再补一个0？不然算出的sa是负的，也就是说j > n

C++ const函数返回值必须为const SprintfWater: @shuideyidi: 嗯，是的。

C++ const函数返回值必须为const 云敛容: 普通的成员变量的get方法不也是const方法嘛，返回值怎么能是const &，你不能以偏概全。你说...

python查找相邻两行为分别包含R\_\_phoenix: 权神居然搞起了python

http://blog.csdn.net/sprintfwater/article/details/9156571

上一篇 POJ 2104(K-th Number-区间第k大-主席树)

下一篇 tarjan算法模板

顶 0 踩 0

主题推荐 算法 存储 struct printf 递归

猜你在找

- 2012年长春赛区小结

最小费用最大流

BZOJ 3876 AHOI2014 支线剧情 费用流

菜鸟都能理解的线段树入门经典

BZOJ-1492-货币兑换cash-NOI2007-CDQ分治
- 【精品课程】C语言基础视频教程

【精品课程】Apple Watch开发入门

【精品课程】VRP系统使用、维护与管理

【精品课程】iOS即时通讯(IM)开发实战篇-基于

【精品课程】Cocos2d-Lua进阶篇（quick）

准备好了么？跳吧！ 更多职位尽在 CSDN JOB

算法和数据挖掘工程师	我要跳槽	计算机视觉算法工程师	我要跳槽
智者四海（北京）技术有限公司	18-35K/月	北京蚁视科技有限公司	10-20K/月
推荐算法工程师	我要跳槽	数据挖掘算法工程师（自然语言方向）	我要跳槽
北京人人车网络技术有限公司	15-20K/月	新浪技术（中国）有限公司	10-20K/月

查看评论

2楼 自在\_飞花 2015-03-12 21:16发表

“值得注意的是，并不是每个root【i】都必须重新开辟size \* log2(size)的空间，”楼主这句话的本意是size×4吧？因为线段树节点的上限是size×4,而不是size×log2(size)。

1楼 hq572241670 2014-08-13 19:42发表

void make\_node(node \*&y,node \*&x,int l,int r,int t) 这里的x,y指针分别是什么意思？

您还没有登录,请[登录]或[注册]

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker OpenStack VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC WAP jQuery BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML LBS Unity Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra CloudStack FTC coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App SpringSide Maemo Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP HBase Pure Solr Angular Cloud Foundry Redis Scala Django Bootstrap

