

q271736642的专栏

目录视图

摘要视图

RSS 订阅

个人资料



xshengh

访问: 4608次

积分: 808分

排名: 第16715名

原创: 77篇

转载: 12篇

译文: 0篇

评论: 1条

文章搜索

文章分类

sicily 动态规划 (1)

sicily 模拟、推理 (6)

Sicily 图论 (1)

美团网面试 (1)

面试 (1)

Leetcode (69)

C/C++ (2)

Oracle数据库 (4)

排序算法 (1)

Ubuntu (2)

Web (1)

文章存档

2013年12月 (18)

2013年11月 (21)

2013年10月 (43)

2013年09月 (7)

阅读排行

2013.9.11 美团网面试题 (643)

Leetcode Maximum Sub (177)

Best Time to Buy and Se (156)

Leetcode Best Time to E (150)

Leetcode Convert Sorter (145)

投票赢好礼，周周有惊喜！

2014年4月微软MVP申请开始了！

消灭0回答，赢下载分

“我的2013”年度征文活动火爆进行中！

专访Kinect手语翻译系统团队

Leetcode Maximum Subarray

分类: Leetcode

2013-10-14 14:12

179人阅读

评论(0)

收藏

举报

分治法

最大子串和

复杂度

Kadane算法

经典最大子串和问题，要求用两种方法实现：1、Kadane算法，复杂度O(n)；2、分治法，复杂度O(nlogn)。

1、Kadane算法：代码比较简单，但是理解需要时间。初始化ans为0，每次用ans加上A[i]的值，并更新最大值，如果遇到ans<0，更新ans的值为0。

[cpp]

01. class Solution {
02. public:
03. int maxSubArray(int A[], int n) {
04. // Note: The Solution object is instantiated only once and is reused by each test
05. int ans = 0, maxn = INT_MIN;
06. for(int i = 0; i < n; i++){
07. if(ans < 0) ans = 0;
08. ans += A[i];
09. maxn = max(maxn, ans);
10. }
11. return maxn;
12. }
13. };

2、分治法：最大子串和的区间有以下三种情况（low，high分别为左右边界，mid为(low+high)/2）：
(1) 区间完全在 A[low,mid-1]
(2) 区间完全在 A[mid+1,high]
(3) 区间包含有 A[mid]
按照这三种情况一直递归下去即可。

[cpp]

01. class Solution {
02. public:
03. int divide(int A[], int low, int high){
04. if(low == high) return A[low];
05. if(low == high-1)
06. return max(A[low]+A[high], max(A[low], A[high]));
07. int mid = (low+high)/2;
08. int lmax = divide(A, low, mid-1);
09. int rmax = divide(A, mid+1, high);
10. int mmax = A[mid];
11. int tmp = mmax;
12. for(int i = mid-1; i >=low; i--){
13. tmp += A[i];
14. if(tmp > mmax) mmax = tmp;
15. }
16. tmp = mmax;
17. for(int i = mid+1; i <= high; i++){
18. tmp += A[i];
19. if(tmp > mmax) mmax = tmp;

bloo.csdn.net/a271736642/article/details/12708291

1/2

Sicily 1028. Hanoi Tower (125)

Leetcode Count and Say (119)

Linux压缩包简体中文乱码 (100)

标准输入流中空白字符的 (80)

Sicily 1001. Alphacode (75)

评论排行

2013.9.11 美团网面试题 (1)

Leetcode Linked List Cycle (0)

经典排序算法 集锦 (0)

Leetcode Remove Duplicates (0)

Leetcode Sort Colors (0)

Leetcode Rotate Image (0)

Linux压缩包简体中文乱码 (0)

Leetcode Populating Next (0)

Leetcode Combinations (0)

Leetcode Search in Rotated (0)

推荐文章

* WebView Attack In Android : 解析第三方账号登录平台所存在的安全隐患

* Android游戏开发十日通 (8) - Box2D的例子和学习小结

* Windows平台Eclipse、MinGW配置OpenCV2.4.4(C++接口调用) 全过程

* Android 双卡双待手机解析短信异常分析及解决

* 为什么使用剪切板时都用GlobalAlloc分配内存

* Android开发学习之ImageView手势拖拽、缩放、旋转

最新评论

2013.9.11 美团网面试题

tilygdx: 这是在华南理工的笔试题吗? 貌似跟哈工大的一样啊

```
20.         }
21.         return max(mmax, max(lmax, rmax));
22.
23.     }
24.     int maxSubArray(int A[], int n) {
25.         // Note: The Solution object is instantiated only once and is reused by each test
26.         return divide(A, 0, n-1);
27.     }
28.
29. };
```

更多 0

上一篇: [Leetcode Populating Next Right Pointers in Each Node](#)

下一篇: [Oracle中TO_DATE、TO_CHAR格式](#)

查看评论

暂无评论

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

专区推荐内容

Android 游戏教程:让人物...
HTML5应用性能调优工具WAP...
Android应用开发
NDK 安卓应用移植方法
Struts2 高危漏洞修复方案
HTML5 经典小游戏之坦克

<< >>

更多招聘职位

我公司职位也要出现在这里

核心技术类目

全部主题

Java VPN Android iOS ERP IE10 Eclipse CRM JavaScript Ubuntu NFC

WAP jQuery 数据库 BI HTML5 Spring Apache Hadoop .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 Spark HBase Pure Solr Angular Cloud Foundry Redis Scala

Django Bootstrap

公司简介 | 招贤纳士 | 广告服务 | 银行汇款帐号 | 联系方式 | 版权声明 | 法律顾问 | 问题报告

QQ客服 微博客服 论坛反馈 联系邮箱: webmaster@csdn.net 服务热线: 400-600-2320

京 ICP 证 070598 号

北京创新乐知信息技术有限公司 版权所有

世纪乐知(北京)网络技术有限公司 提供技术支持

江苏乐知网络技术有限公司 提供商务支持

Copyright © 1999-2012, CSDN.NET, All Rights Reserved 

blog.csdn.net/q271736642/article/details/12708291

2/2