

ju136的专栏

个人资料



ju136

访问：39168次

积分：1135分

排名：第9932名

原创：76篇

转载：3篇

译文：0篇

评论：55条

文章搜索

文章分类

OpenCV (2)

MIT-6.828 (1)

算法导论 (1)

Linux内核0.11 (0)

云计算-OpenStack (2)

汇编语言-gas (2)

ACM (25)

Linux工具类 (4)

编程珠玑 (21)

数据结构算法面试题精选及整理 (5)

具体数学 (0)

面试 (17)

The C Programming Language (1)

文章存档

2013年08月 (1)

2013年03月 (1)

2012年11月 (4)

2012年10月 (10)

2012年08月 (1)

展开

阅读排行

CSDN个性阅读，订阅你想看的博客文章

专访李铁军：从医生到金山首席安全专家的转变

独一无二的职位：开源社区经理

CSDN博客第四期移动开发最佳博主评选

与她合影留念，赢IT图书

编程珠玑第一章

分类：编程珠玑

2011-09-29 00:36

610人阅读

评论(0)

收藏

举报

编程

电话

c

fun

测试

存储

原文中的问题：

如何在1MB的空间里面对一千万个整数进行排序？并且每个数都小于1千万。实际上这个需要1.25MB的内存空间。

1MB总共有838, 8608。所以估计也可以在1MB左右的空间里面进行排序了。

【cpp】

```
01. #include <stdio.h>
02. #include <stdlib.h>
03. #define BITSPERWORD 32
04. #define SHIFT 5
05. #define MASK 0x1F
06. #define N 10000000
07. int a[1 + N/BITSPERWORD];
08. void set(int i) { a[i>>SHIFT] |= (1<<(i & MASK)); }
09. void clr(int i) { a[i>>SHIFT] &= ~(1<<(i & MASK)); }
10. int test(int i){ return a[i>>SHIFT] & (1<<(i & MASK)); }
11. int main()
12. {
13.     int i = 0, top = 1 + N/BITSPERWORD;
14.     memset(a, 0, sizeof(a)*sizeof(int));
15.     while (scanf("%d", &i) != EOF) set(i);
16.     for (i = 0; i < N; i++)
17.         if (test(i)) printf("%d\n", i);
18.     return 0;
}
```

课后的题目：

1、使用库来进行排序

【cpp】

```
01. #include <stdio.h>
02. #include <stdlib.h>
03. #define ms 1025
04. int a[ms];
05. int cmp(const void *a, const void *b)
06. {
07.     return (*(int*)a) - (*(int*)b);
08. }
```

面试题整理-最小没出现的整数	(2063)
数据结构算法面试题精选	(1465)
HDU OJ 题目分类	(1323)
数据结构算法面试题精选	(1224)
OpenCV源码阅读(1)	(1101)
OpenCV源码阅读(2)	(1013)
编程珠玑第三章	(824)
MIT-6.828-lab1.1	(762)
POJ 1205	(727)
编程珠玑第二章	(724)

评论排行

面试题整理-最小没出现的整数	(11)
数据结构算法面试题精选	(10)
编程珠玑第五章	(5)
编程珠玑第9章	(5)
编程珠玑第8章	(4)
ACM 算法艺术与信息学竞赛	(3)
编程珠玑第二章	(3)
编程珠玑第8章一课后习题	(2)
ACM 算法艺术与信息学竞赛	(2)
编程珠玑第12章	(2)

推荐文章

\* 《C++游戏开发》十八角色在障碍物中智能行走的实现

\* 百度地图之路线规划

\* Android 仿PhotoShop调色板应用(三) 主体界面绘制

\* javascript特效：会随着鼠标而动的眼睛

\* Android混合开发之Activity类与html页面之间的相互跳转(并解决黑屏问题)

\* cocos2d-x 通过JNI实现c/c++和lua的交互

最新评论

编程珠玑第8章  
tplee923: 第十题求出sum数组后你如何在n!gn之内求出两个数的绝对值最接近于某一个数呢？我知道如果...

MySQL用户添加  
ju136: 最好的处理方法是让MySQL单独位于某台机器。

面试题整理-空格变更  
ju136: @linuxlinuxlinuxlinux: 对哦，应该返回NULL吧。

面试题整理-最小没出现的整数  
ju136: @fangjuntan: 看来你还没看懂~

面试题整理-空格变更  
linuxlinuxlinuxlinux: 第九行该返回什么啊 if (!src) return;

面试题整理-最小没出现的整数  
super\_wlb: 介绍很仔细，不错

面试题整理-最小没出现的整数  
ju136: @fangjuntan: 不会的

ACM 算法艺术与信息学竞赛 1.2.1\_binbin: Orz，代码这么简练，

面试题整理-最小没出现的整数  
fangjuntan: @fangjuntan: 这个while应该改为if。。。

面试题整理-最小没出现的整数  
fangjuntan: while (0 < a && a < n && i != a) {t = a;...

```
09.     int main(void)
10.     {
11.         int n, i;
12.         while (scanf("%d", &n) != EOF)
13.         {
14.             for (i = 0; i < n; ++i) scanf("%d", &a[i]);
15.             qsort(a, n, sizeof(int), cmp);
16.             for (i = 0; i < n - 1; ++i) printf("%d ", a[i]);
17.             printf("%d\n", a[i]);
18.         }
19.         return 0;
20.     }
```

2、使用位运算

```
[cpp]

01.     void set(int i) { a[i>>SHIFT] |= (1<<(i & MASK)); }
02.     void clr(int i) { a[i>>SHIFT] &= ~(1<<(i & MASK)); }
03.     int test(int i) { return a[i>>SHIFT] & (1<<(i & MASK)); }
```

3、比较位图排序与系统排序

位图排序是最快的，针对这个问题而言，qsort比stl sort速度快。

4、随机生成[0, n)之间不重复的随机数

```
[cpp]

01.     for (i = 0; i < n; ++i)
02.         a[i] = i;
03.     for (i = 0; i < n; ++i)
04.     {
05.         pos = rand()%(n - i) + i;
06.         t = a[i]; a[i] = a[pos]; a[pos] = t;
07.     }
```

5、如果1MB是严格控制的空间，如果数据有1.25MB的bit数目。那么应该是需要读取2次。

k = 需要跑几趟直接用需要排序的数据量/内存空间bit数，往上取整则可。

时间开销 = kn

空间开销 n/k

注意的是，每次在扫描的时候，取数据的范围是不一样的。

6、如果每个数据出现最多10次，那么需要4个bit位来刻录一个数。这时存储空间减小至原来的1/4。

那么如果一定要按照bitmap的方式来进行处理，则需要利用5题中的结论。

7、问题：[R. Weil]本书1.4节中描述的程序存在一些缺陷。首先是假定在输入中没有出现两次的整数。如果某个数出现超过一次的话，会发生什么？在这种情况下，如何修改程序来调用错误处理函数？当输入整数小于零或大于等于n时，又会发生什么？如果某个输入不是数值又如何？在这些情况下，程序该如何处理？程序还应该包含哪些明智的检查？描述一些用以测试程序的小型数据集，并说明如何正确处理上述以及其他不良情况。

如果某个数出现超过一次的话，会发生什么？

会被忽略掉，因为原来的程序本身就是用来处理只出现一次的情况的。

在这种情况下，如何修改程序来调用错误处理函数？

```
[html]

01.     while (scanf("%d", &i) != EOF)
02.         if (test(i)) call_error_fun();
03.         else set(i);
```

当输入整数小于零或大于等于n时，又会发生什么？

会出现访问越界的情况。-1访问时，会访问a[-1]的31个bit位。

如果某个输入不是数值又如何？在这些情况下，程序该如何处理？

输入可能是浮点数，或是字符什么的~~

可以先读入字符串，再用atoi转换为整形数，如果失败，则进行出错处理。

程序还应该包含哪些明智的检查？

8、免费电话号码至少有800，878，888等，那么如何查看一个号码是否是免费号码。？

第一种方案：如果是一千万个电话号码都有可能成为免费号码，那么至需要1.25MB \* (免费号码前缀个数)。

第二种方案：省空间，多次扫描文件：

1、首先扫描整个文件，看有哪个免费号码前缀。以及每个免费号码前缀下的号码个数。

2、设置区间映射表：比如800前缀有125个免费号码，找到最大的数，与最小的数，差值做为bit长度。

**第三种方案：建立索引的方式来进行处理。以最后7位为索引，后面800,878什么的，为值。如果不是免费号码，应该是不用加入到这个hash表中。**

9、避免初始化问题

做法是：使用两个等长的辅助数组，比如要把a[n]初始化，那么在第一次访问时：

b[i] = top;

c[b[i]] = i;

++top;

给出示例代码

```
[html]
01.  #include<stdio.h>
02.  #include<stdlib.h>
03.  #include<string.h>
04.
05.  #define ms 100
06.  int a[ms];
07.  int b[ms];
08.  int c[ms];
09.  int top;
10.
11.  //判断是否被初始化过。
12.  bool is_init(int i)
13.  {
14.      return (b[i] < top && c[b[i]] == i);
15.  }
16.  int main(void)
17.  {
18.      top = 0;
19.      //这里生成一些随机数值。
20.      for (int i = 0; i < ms; ++i)
21.          a[i] = b[i] = c[i] = i;
22.      for (int i = 0; i < ms; ++i)
23.      {
24.          if (is_init(i))
25.          {
26.              printf("error");
27.          }
28.          int v = i + rand()%(ms - i + 1);
29.          int t = a[i]; a[i] = a[v]; a[v] = t;
```

```
30.
31.         v = i + rand()%(ms - i + 1);
32.         t = b[i]; b[i] = b[v]; b[v] = t;
33.
34.         v = i + rand()%(ms - i + 1);
35.         t = c[i]; c[i] = c[v] ; c[v] = t;
36.     }
37.
38.     for (int i = 0; i < ms; ++i)
39.     {
40.         if (is_init(i) == false)
41.         {
42.             a[i] = i;
43.             b[i] = top;
44.             c[top++] = i;
45.         }
46.     }
47.
48.     for (int i = 0; i < ms; ++i)
49.     {
50.         if (!is_init(i))
51.         {
52.             printf("error: %d\n", i);
53.         }
54.     }
55.     return 0;
56. }
```

10、由于唯一性，采用hash啊。

上一篇：[【汇编学习】第二章：程序起步](#)

分享到：

下一篇：[编程珠玑第二章](#)

Visual Studio Premium With MSDN  
Surface Pro 128G含键盘  
Xamarin跨平台开发工具

原价¥83,588 现在只要 **¥61,800**  
仅限CSDN企业会员购买！

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

- 我的HTML5学习经历
- 利用触摸屏的实时战略游戏
- 将 Win32 应用移植至 Wi...
- Android 开发之多线程处理...
- 跟燕青一起学Windows8应用...
- 实现高性能 gsoap 服务

<< >>

更多招聘职位

我公司职位也要出现在这里

- 【同方鼎欣信息技术有限公司】测试工程师
- 【卓望信息技术（北京）有限公司】测试工程师
- 【载信软件（上海）有限公司】测试工程师
- 【广东南方海岸科技服务有限公司】高级测试工程师
- 【深圳矽递科技有限公司】硬件工程师
- 【杭州派尔科技】Winform开发工程师