

# 可笑痴狂

[博客园](#) [首页](#) [博问](#) [闪存](#) [新随笔](#) [订阅](#) [管理](#)

随笔 - 390, 文章 - 34, 评论 - 62, 阅读 - 132万

## 基于贪心算法的几类区间覆盖问题

### 基于贪心算法的几类区间覆盖问题：

#### (1) 区间完全覆盖问题

问题描述：

给定一个长度为m的区间，再给出n条线段的起点和终点（注意这里是闭区间），求最少使用多少条线段可以将整个区间完全覆盖

样例：

区间长度8，可选的覆盖线段[2,6],[1,4],[3,6],[3,7],[6,8],[2,4],[3,5]

解题过程：

1、将每一个区间按照左端点递增顺序排列，拍完序后为[1,4], [2,4], [2,6], [3,5],

[3,6], [3,7], [6,8]

2、设置一个变量表示已经覆盖到的区域。再剩下的线段中找出所有左端点小于等于当前

已经覆盖到的区域的右端点的线段中，右端点最大的线段在加入，直到已经覆盖全部的区域

3、过程：

假设第一步加入[1,4]，那么下一步能够选择的有[2,6], [3,5], [3,6], [3,7]，

由于7最大，所以下一步选择[3,7]，最后一步只能选择[6,8]，这个时候刚好

达到了8退出，所选区间为3

4、贪心证明：

需要最少的线段进行覆盖，那么选取的线段必然要尽量长，而已经覆盖到的区域之前



昵称：[可笑痴狂](#)

园龄：[9年4个月](#)

粉丝：[244](#)

关注：[35](#)

[+加关注](#)

$\leq$	2013年4月						$\geq$
日	一	二	三	四	五	六	
31	1	2	3	4	5	6	
7	8	9	10	11	12	13	
14	15	16	17	18	19	20	
21	22	23	24	25	26	27	
28	29	30	1	2	3	4	
5	6	7	8	9	10	11	

搜索

找找看

谷歌搜索

最新随笔

的地方已经无所谓了，（可以理解成所有的可以覆盖的左端点都是已经覆盖到的地方），那么真正能够使得线段更成的是右端点，左端点没有太大的意义，所以选择右端点来覆盖

## （2）最大不相交覆盖(我总感觉这个算法不对，这不应该和会议安排问题一样吗？ 直接按照终点排序再依次选择？？？)

问题描述:

给定一个长度为m的区间，再给出n条线段的起点和终点（开区间和闭区间处理的方法是不同，这里以开区间为例），问题是从中选取尽量多的线段，使得每个线段都是独立的，就是不和其它有任何线段有相交的地方

样例:

区间长度8，可选的覆盖线段[2,6],[1,4],[3,6],[3,7],[6,8],[2,4],[3,5]

解题过程:

对线段的右端点进行升序排序，每加入一个线段，然后选择后面若干个

（也有可能是一个）右端点相同的线段，选择左端点最大的那一条，如果加入以后不会跟之前的线段产生公共部分，那么就加入，否则就继续判断后面的线段

1、排序:将每一个区间按右端点进行递增顺序排列，拍完序后为[1,4], [2,4], [3,5], [2,6], [3,6], [3,7], [6,8]

2、第一步选取[2,4]，发现后面只能加入[6,8],所以区间的个数为2

3、贪心证明:因为需要尽量多的独立的线段，所以每个线段都尽可能的小，

对于同一右端点，左端点越大，线段长度越小。那么为什么要对右端点进行排序呢？

如果左端点进行排序，那么右端点是多少并不知道，那么每一条线段都不能对之前所有的线段进行一个总结，那么这就明显不满足贪心的最有字结构了。

## （3）区间选点问题

问题描述:

给定一个长度为m的区间，再给出n条线段和这n条线段需要满足的要求

（要求是这n条线段上至少有的被选择的点的个数），问题是整个区间内最少选择几个点，使其满足每一条线段的要求。

样例: 略

解题过程:

将每个线段按照终点坐标进行递增排序，相同终点的前点坐标从大到小排列，一个将其满足（每次选择的点为该条线段的右端点）

贪心证明:

要想使得剩下的线段上选择的点最少，那么就应该尽量使得已经选择了的点尽量能在后面的线段中发挥作用，而我们是从小往右选择线段的，那么要使得选取的点能满足后面线段的要求，那么必须是从线段的右端点开始选点，那么问题(2)一样涉及到一个问题，如果是按照线段的左端点对线段进行排序的话，不知道右端点的话，每一条线段都不能对之前已经操作过的所有线段进行一个总结，那么这就同样不满足贪心算法的最优子结构性质了。

[1.批处理实现批量创建快捷方式](#)

[2.设计模式（六）装饰模式（转）](#)

[3.设计模式（五）桥接模式（转）](#)

[4.设计模式（四）适配器模式（转）](#)

[5.设计模式（三）建造者模式（转）](#)

[6.设计模式（二）单例模式（转）](#)

[7.设计模式（一）工厂模式（转）](#)

[8.读者写者问题](#)

[9.生产者消费者问题](#)

[10.找出数组中出现次数超过一半的元素](#)

## 积分与排名

积分 - 484458

排名 - 988

## 随笔分类 (487)

[C++\(44\)](#)

[hash\(3\)](#)

[java学习\(4\)](#)

[KMP算法\(6\)](#)

[MFC\(23\)](#)

[PHP\(6\)](#)

[STL\(20\)](#)

[TCP/IP\(3\)](#)

[Windows编程\(9\)](#)

[背包问题\(13\)](#)

[并查集\(14\)](#)

[策略/博弈\(37\)](#)

[大数/高精度\(9\)](#)

[递归/递推\(10\)](#)

[动态规划\(22\)](#)

[更多](#)

## 随笔档案 (389)

可以解决的实际问题：数轴上面有n个闭区间[a,b],取尽量少的点，使得每个区间内都至少有一个点（不同区间内含的点可以是同一个）

应用例题：（貌似不是很简单。。。）

有一列整数，他的每一个数各不相同，我们不知道有多少个，但我们知道在某些区间中至少有多少个整数，用区间（L，R，C）来描述，表示整数序列中至少有C个整数来自子区间[L, R]，若干个这样的区间，问这个整数序列的长度最少能为多少。

区间选点算法实现：



```
1 #include <iostream>
2 #include <algorithm>
3
4 using namespace std;
5
6 struct line
7 {
8     int left;
9     int right;
10 }a[100];
11
12 bool cmp(line p, line q)
13 {
14     if(p.right != q.right)
15         return p.right < q.right;
16     return p.left > q.left;
17 }
18
19 int main()
20 {
21     int n;
22     while(cin >> n)
23     {
24         for(int i = 0; i < n; ++i)
```

[2016年4月\(1\)](#)  
[2016年3月\(1\)](#)  
[2016年2月\(3\)](#)  
[2016年1月\(2\)](#)  
[2015年12月\(4\)](#)  
[2015年11月\(1\)](#)  
[2015年5月\(2\)](#)  
[2015年4月\(6\)](#)  
[2015年3月\(1\)](#)  
[2015年1月\(1\)](#)  
[2014年12月\(6\)](#)  
[2014年11月\(1\)](#)  
[2014年10月\(2\)](#)  
[2014年9月\(2\)](#)  
[2014年8月\(1\)](#)  
[更多](#)

文章分类 (34)

[c++\(11\)](#)  
[hash函数\(1\)](#)  
[STL学习指南\(6\)](#)  
[各种应用\(7\)](#)  
[模式匹配----KMP\(2\)](#)  
[排序\(2\)](#)  
[树状数组\(1\)](#)  
[位运算\(1\)](#)  
[线段树\(2\)](#)  
[最小生成树和次小生成树\(1\)](#)

文章档案 (34)

[2013年9月\(1\)](#)  
[2013年7月\(1\)](#)  
[2013年3月\(1\)](#)  
[2012年12月\(2\)](#)  
[2012年11月\(11\)](#)

```

25         cin >> a[i].left >> a[i].right;
26         sort(a, a + n, cmp);
27         int cnt = 0;
28         int end = -1;
29         for(int i = 0; i < n; ++i)
30         {
31             if(end >= a[i].left && end <= a[i].right)
32                 continue;
33             else
34             {
35                 ++cnt;
36                 end = a[i].right;
37             }
38         }
39         cout << cnt << endl;
40     }
41     return 0;
42 }

```



功不成，身已退

分类: [贪心](#)

标签: [基于贪心算法的几类区间覆盖问题](#), [区间覆盖](#)

好文要顶

关注我

收藏该文



可笑痴狂

关注 - 35

粉丝 - 244

+加关注

« 上一篇: [POJ-2186 Popular Cows \(强连通缩点\)](#)

» 下一篇: [最长公共递增子序列 \(LCIS\)](#)

0

1

[2012年10月\(2\)](#)

[2012年9月\(1\)](#)

[2012年8月\(12\)](#)

[2012年7月\(2\)](#)

[2012年6月\(1\)](#)

## 博客链接

[三江小渡](#)

[算法天才](#)

[先前新浪博客](#)

[开开甲](#)

[知行](#)

[算法专题](#)

[算法导论学习博客](#)

## 阅读排行榜

1. [Oracle中Merge into用法总结\(200392\)](#)

2. [在VS中添加lib库的三种方法\(80675\)](#)

3. [C++抽象类\(39267\)](#)

4. [Windows共享内存示例\(35688\)](#)

5. [不能从const char \\*转换为LPCWSTR --VS经常碰到\(31717\)](#)

## 评论排行榜

1. [Oracle中Merge into用法总结\(5\)](#)

2. [HDU 1869 六度分离 ----Floyd\(5\)](#)

3. [NYOJ-68 三点顺序 --有向面积\(4\)](#)

4. [第一个手写Win32窗口程序\(3\)](#)

5. [C++抽象类\(3\)](#)

## 推荐排行榜

1. [Oracle中Merge into用法总结\(13\)](#)

2. [POJ-1664 放苹果\(7\)](#)

3. [在VS中添加lib库的三种方法\(5\)](#)