登录 | 注册 登录 | 注册

## **Evil.livE**

:= 目录视图 ₩ 摘要视图 RSS 订阅 Android 6.0 运行时权限处理完全解析 python编程常用模板总结 【知识库】一张大图看懂Android架构

### 【计算几何】zjoi2008 risk

标签: output go input c math 算法

2011-02-25 20:30 评论(0) 临端 举报 1316人阅读

■版权声明:本文为博主原创文章,未经博主允许不得转载。

题目大意:给你一个平面图,平面被分成了若干个区域,求出每个区域与之相邻的区域,数据保证区域数不超过500;

有数据范围可以看出要求是低于N<sup>3</sup>的算法。

#### 我的做法是:

1.将每一条线段AB拆成两条有向线段AB与BA,然后将所有有向线段AB与从B点引出的线段中位于AB顺时针方向的第一条线段连有向边(BA与AB的夹 角为2PI,而不是0)。这样对所有的线段做一次DFS之后就能将图中的封闭区域全部求出来,这样出了某个区域包含的区域,其他相邻的区域都可以求出 来了;

2.可以使用环顾法求出各个区域之间的包含关系,这个包含关系类似于一棵树,然后用拓扑序,先处理叶子节点,这样可以把包含关系也求出来。

这个方法是我觉得最好实现的方法了。

总之既然是计算几何题就肯定有很多细节的,最后还是拿了几组数据来调才A的(数据很不好生成。。)

另外就是发现math库里面直接就有求极角的函数了,那就是arctan2!以后不用自己写极角了,每次都要重新推一遍,囧。

#### code:

```
[delphi]
01.
      program ex3;
02.
      uses math;
03.
      type xl=record x,y:longint end;
04.
05.
       h,dp:array[0..5000] of xl;
        jd:array[0..10000] of real;
06.
        c,e,a:array[0..600,0..600] of boolean;
07.
        d,p,b,g,s,v,q,go,fo,fir,next:array[-1..10000] of longint;
08.
09.
        t,n,m,i,j,k,l,dt,mt,time:longint;
        ss,tt,t1:xl;
10.
11.
        t2:real;
12.
13.
      operator /(a,b:xl)c:longint;begin c:=a.x*b.x+a.y*b.y end;
      operator *(a,b:xl)c:longint;begin c:=a.x*b.y-a.y*b.x end;
14.
15.
      function mo(a:xl):real;begin mo:=sqrt(a.x*a.x+a.y*a.y) end;
      operator -(a,b:xl)c:xl;begin c.x:=a.x-b.x;c.y:=a.y-b.y end;
16.
      function jiao(a,b:xl):real;
17.
```

File failed to load: file:///C:/Users/wkl31415926/OneDrive/IO wkl /%E5%85%A5%E9%97%A8%E9%A2%98%E7%9B%AE%E6%95%B4%E7%90%86/%E5%87%A0%E4%BD%95/%

Jiao:=(a/p)/(mo(a)\*mo(p));

```
if jiao=1 then jiao:=0 else jiao:=arccos(jiao);
         21.
                         if a*b<0 then jiao:=-jiao;</pre>
         22.
         23.
                      function find(dd:xl):longint;var i:longint;
         24.
         25.
                          for i:=1 to dt do
         26.
                              if (dd.x=dp[i].x)and(dd.y=dp[i].y) then exit(i);
         27.
         28.
                          inc(dt);dp[dt]:=dd;find:=dt;
         29.
         30.
                      procedure link(i,j:longint);
         31.
         32.
                         inc(t);next[t]:=d[i];d[i]:=t;p[t]:=j;b[t]:=t+1;
         33.
         34.
                         inc(t);next[t]:=d[j];d[j]:=t;p[t]:=i;b[t]:=t-1;
         35.
                      end:
         36.
         37.
                      begin
                         assign(input,'risk.in');reset(input);
         38.
         39.
                          assign(output,'risk.out');rewrite(output);
         40.
         41.
                          for i:=1 to n do readln(h[i].x,h[i].y);
         42.
                          for i:=1 to m do begin
         43.
                              readln(ss.x,ss.y,tt.x,tt.y);
         44.
                              link(find(ss),find(tt));
         45.
                          end:
         46.
                          for i:=1 to t do begin
                             jd[i]:=-<mark>9</mark>;
         47.
         48.
                              k:=d[p[i]];j:=p[k];
         49.
                              t1{:=}dp[p[i]]{-}dp[p[b[i]]];
         50.
                              while k<>0 do begin
                                 if k<>b[i] then begin
         51.
         52.
                                       t2:=jiao(t1,dp[j]-dp[p[i]]);
                                      if t2>jd[i] then begin
         53.
         54.
                                          jd[i]:=t2;
         55.
                                           go[i]:=k;
         56.
                                       end;
         57.
                                   end:
         58.
                                   k:=next[k];j:=p[k];
         59.
                              end:
         60.
                          for i:=1 to t do
         61.
         62.
                              if fo[i]=0 then begin
         63.
                                  k:=i;
         64.
                                   t2:=<mark>0</mark>;
         65.
                                  inc(mt);
         66.
                                   v[mt]:=0;
         67.
                                  repeat
         68.
                                      t2:=t2+jd[k];
         69.
                                     inc(v[mt]);
         70.
                                      k:=go[k];
         71.
                                   until i=k;
         72.
                                  if t2>0 then begin
         73.
                                      repeat
         74.
                                          fo[k]:=mt;
         75.
                                          k:=go[k];
         76.
                                       until i=k:
         77.
                                      fir[mt]:=i;
         78.
                                   end else begin
         79.
                                      repeat
         80.
                                           fo[k]:=-1;
         81.
                                           k:=go[k];
         82.
                                       until i=k;
         83.
                                       dec(mt);
         84.
                                   end;
         85.
                              end;
         86.
                          for i:=1 to n do
         87.
                              for j:=1 to mt do begin
         88.
                                 k:=fir[j];
         89.
         90.
                                  repeat
         91.
                                      t2:=t2+jiao(dp[p[b[k]]]-h[i],
         92.
                                                                 dp[p[k]]-h[i]);
         93.
                                      k:=go[k];
         94.
                                   until k=fir[j];
         95.
                                   c[i,j]:=abs(t2-2*pi)<1e-6;
         96.
                                   inc(g[j],ord(c[i,j]));
         97.
                                   a[i,j]:=c[i,j];
File failed to load: file:///C:/Users/wkl31415926/OneDrive/IO_wkl_/%E5%85%A5%E9%97%A8%E9%A2%98%E7%9B%AE%E6%95%B4%E7%90%86/%E5%87%A0%E4%BD%95/%E5%E9%P7%A8%E9%A2%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8%E7%P8
                         for 1:=1 to n do begin
```

```
100.
                           for i:=1 to mt do
      101.
                              if g[i]=1 then break;
      102.
                           for j:=1 to n do
      103.
                             if c[j,i] then break;
      104.
                           s[i]:=j;q[j]:=i;
      105.
                           for k:=1 to mt do dec(g[k],ord(c[j,k]));
      106.
                           for k:=1 to mt do c[j,k]:=false;
      107.
                           for j:=1 to n do
      108.
                             if (j<>s[i])and a[j,i] and (v[q[j]]<>0) then begin
      109.
                                  v[q[j]]:=0;
      110.
                                  e[s[i],j]:=true;
      111.
                                  e[j,s[i]]:=true;
      112.
                              end;
      113.
                           k:=fir[i];
      114.
                          repeat
                              dec(v[fo[b[k]]]);
      115.
      116.
                              k:=go[k];
      117.
                           until k=fir[i];
      118.
      119.
                       for i:=1 to mt do begin
      120.
                          k:=fir[i];
      121.
                           repeat
                              e[s[fo[b[k]]],s[i]]:=true;
      122.
      123.
                              k:=go[k];
      124.
                           until k=fir[i];
      125.
                       for i:=1 to n do begin
      126.
      127.
                         t:=0;
      128.
                           for j:=1 to n do inc(t,ord(e[i,j]));
                          write(t,' ');
      129.
                         for j:=1 to n do
      130.
                             if e[i,j] then write(j,' ');
      131.
      132.
      133.
      134.
                       close(input);close(output);
      135.
                  end.
                   [delphi]
        01.
        02.
                   uses math:
        03.
                   type xl=record x,y:longint end;
        04.
        05.
                       h,dp:array[0..5000] of xl;
        06.
                       jd:array[0..10000] of real;
        07.
                       c,e,a:array[0..600,0..600] of boolean;
                       d,p,b,g,s,v,q,go,fo,fir,next:array[-1..10000] of longint;
        08.
        09.
                       t,n,m,i,j,k,l,dt,mt,time:longint;
        10.
                       ss,tt,t1:xl;
        11.
                       t2:real;
        12.
        13.
                   operator /(a,b:xl)c:longint;begin c:=a.x*b.x+a.y*b.y end;
                   operator *(a,b:xl)c:longint;begin c:=a.x*b.y-a.y*b.x end;
        14.
        15.
                   function mo(a:xl):real;begin mo:=sqrt(a.x*a.x+a.y*a.y) end;
        16.
                   operator -(a,b:xl)c:xl;begin c.x:=a.x-b.x;c.y:=a.y-b.y end;
        17.
                   function jiao(a,b:xl):real;
        18.
                   begin
        19.
                       jiao:=(a/b)/(mo(a)*mo(b));
        20.
                       if jiao=1 then jiao:=0 else jiao:=arccos(jiao);
        21.
                      if a*b<0 then jiao:=-jiao;</pre>
        22.
        23.
        24.
                   function find(dd:xl):longint;var i:longint;
        25.
        26.
                       for i:=1 to dt do
        27.
                           if (dd.x=dp[i].x)and(dd.y=dp[i].y) then exit(i);
        28.
                       inc(dt);dp[dt]:=dd;find:=dt;
        29.
                   end;
        30.
        31.
                   procedure link(i,j:longint);
        32.
        33.
                       inc(t);next[t]:=d[i];d[i]:=t;p[t]:=j;b[t]:=t+1;
        34.
                       inc(t);next[t]:=d[j];d[j]:=t;p[t]:=i;b[t]:=t-1;
        35.
                   end:
        36.
        37.
                   begin
                       assign(input,'risk.in');reset(input);
                       assign(output,'risk.out');rewrite(output);
File failed to load: file:///C:/Users/wkl31415926/OneDrive/IO_wkl_/%E5%85%A5%E9%97%A8%E9%A2%98%E7%9B%AE%E6%95%B4%E7%90%86/%E5%87%A0%E4%BD%95/%E5%E9%P1%E6%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6%P1%E6
                      for i:=1 to n do readln(h[i].x,h[i].y);
```

```
42.
         for i:=1 to m do begin
 43.
           readln(ss.x,ss.y,tt.x,tt.y);
 44.
           link(find(ss),find(tt));
 45.
         end;
 46.
         for i:=1 to t do begin
 47.
           jd[i]:=-9;
 48.
           k:=d[p[i]];j:=p[k];
 49.
           t1:=dp[p[i]]-dp[p[b[i]]];
 50.
           while k<>0 do begin
 51.
             if k<>b[i] then begin
 52.
               t2:=jiao(t1,dp[j]-dp[p[i]]);
 53.
               if t2>jd[i] then begin
 54.
                 jd[i]:=t2;
 55.
                 go[i]:=k;
               end;
 57.
             end;
 58.
             k:=next[k];j:=p[k];
 59.
           end;
 60.
 61.
         for i:=1 to t do
 62.
           if fo[i]=0 then begin
 63.
             k:=i:
 64.
             t2:=<mark>0</mark>;
 65.
             inc(mt);
 66.
             v[mt]:=0;
 67.
             repeat
 68.
               t2:=t2+jd[k];
 69.
               inc(v[mt]);
 70.
               k:=go[k];
 71.
             until i=k:
 72.
             if t2>0 then begin
 73.
               repeat
 74.
                 fo[k]:=mt;
 75.
                 k:=go[k];
 76.
               until i=k;
 77.
               fir[mt]:=i;
 78.
             end else begin
 79.
               repeat
 80.
                 fo[k]:=-1;
 81.
                 k:=go[k];
 82.
               until i=k;
               dec(mt);
 83.
 84.
             end;
 85.
           end;
 86.
         for i:=1 to n do
 87.
           for j:=1 to mt do begin
 88.
             k:=fir[j];
 89.
             t2:=0;
 90.
             repeat
               t2:=t2+jiao(dp[p[b[k]]]-h[i],
 91.
 92.
                            dp[p[k]]-h[i]);
 93.
               k:=go[k];
             until k=fir[j];
 94.
 95.
             c[i,j]:=abs(t2-2*pi)<1e-6;
 96.
             inc(g[j],ord(c[i,j]));
 97.
             a[i,j]:=c[i,j];
 98.
           end;
         for 1:=1 to n do begin
 99.
100.
           for i:=1 to mt do
101.
             if g[i]=1 then break;
102.
           for j:=1 to n do
103.
            if c[j,i] then break;
104.
           s[i]:=j;q[j]:=i;
105.
           for k:=1 to mt do dec(g[k],ord(c[j,k]));
106.
           for k:=1 to mt do c[j,k]:=false;
107.
           for j:=1 to n do
108.
             if (j <> s[i]) and a[j,i] and (v[q[j]] <> 0) then begin
109.
               v[q[j]]:=0;
110.
               e[s[i],j]:=true;
111.
               e[j,s[i]]:=true;
112.
             end;
113.
           k:=fir[i];
114.
           repeat
115.
             dec(v[fo[b[k]]]);
116.
             k:=go[k];
117.
           until k=fir[i];
118.
         for i:=1 to mt do begin
```

File failed to load: file:///C:/Users/wkl31415926/OneDrive/IO\_wkl\_/%E5%85%A5%E9%97%A8%E9%A2%98%E7%9B%AE%E6%95%B4%E7%90%86/%E5%87%A0%E4%BD%95/%

121.

```
122.
            e[s[fo[b[k]]],s[i]]:=true;
123.
            k:=go[k];
          until k=fir[i];
124.
125.
        end;
126.
        for i:=1 to n do begin
127.
          t:=0;
          for j:=1 to n do inc(t,ord(e[i,j]));
128.
          write(t,' ');
129.
130.
          for j:=1 to n do
            if e[i,j] then write(j,' ');
131.
132.
          writeln;
133.
        end:
134.
        close(input);close(output);
135. end.
```

顶

上一篇 最近做的几道恶心题

下一篇 带标记非递归的动态树

#### 我的同类文章

```
习题总结(33)
                                  更多文章
```

#### 猜你在找

```
《C语言/C++学习指南》加密解密篇(安全相关算法)
C语言系列之 递归算法示例与 Windows 趣味小项目
C语言系列之 字符串相关算法
C语言系列之 快速排序与全排列算法
C语言系列之 字符串压缩算法与结构体初探
2017软考软件设计师下午案例分析视频培训课程
2017软考系统集成项目管理工程师视频教程 案例分析精讲
2017软考系统集成项目管理工程师-上午历年真题解析培训视频课程
MATLAB基础知识大串讲
走进Vue.js 1.0
```

#### 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

#### 个人资料



File failed to load: file:///C:/Users/wkl31415926/OneDrive/IO\_wkl\_/%E5%85%A5%E9%97%A8%E9%A2%98%E7%9B%AE%E6%95%B4%E7%90%86/%E5%87%A0%E4%BD%95/%E5%E9%P1%E6%E6%P1%E6

```
。 访问: 121524次
。 积分: 2043
。 等级: BLDG > 5
。排名:第11652名
    。 原创:69 转载:0篇
           。 译文:0篇
    。 评论:135
```

文章搜索		

# 文章分类 习题总结(34) 基本算法(21)

# 文章存档 2011年10月(5) 2011年09月(3) 2011年08月(1) 2011年07月(1) 2011年06月(1) 展开

```
阅读排行
【动态树初探】link-cut
tree(13042)
【费用流算法: ZKW】(4284)
【快速傅里叶变换】FFT高精乘
法(3867)
【划分树】求区间K大数(3629)
【带除法的取模运算】hnoi2009
有趣的数列(3190)
【计算几何】圆的面积并(2949)
【连通性状态压缩DP】
poj1739(2650)
OJ记录 (11.4.8) (2524)
单旋SPLAY的优化(2512)
【noi2005】维护数列(2474)
```

```
评论排行
【动态树初探】link-cut tree(16)
单旋SPLAY的优化(10)
光线追踪——最终版(8)
【计算几何】圆的面积并(7)
【noi2005】维护数列(7)
【可并堆的应用】poj3016(6)
【容斥原理+状态压缩】zjoi2009
多米诺骨牌(5)
【连通性状态压缩DP】
poj1739(5)
```

组合计数小启发(4)

#### 推荐文章

- 。 \*Android自定义 ViewGroup打造各种风格 的SlidingMenu
- 。 \* Android 6.0 运行时权限 处理完全解析
- 。 \* 数据库性能优化之SQL 语句优化
- 。 \*Animation动画详解(七) –ObjectAnimator基本 使用
- 。 \* Chromium网页URL加载 过程分析
- \* 大数据三种典型云服务 模式

#### 最新评论

【动态树初探】link-cut tree no\_ip: 难道这就是传说中trajan的 spaly...

【计算几何】圆的面积并 estivalswan: 求问sort函数里面 t1:=r; 这个>>是什么符号呀?

【计算几何】圆的面积并 estivalswan: 你好这篇的原论文 是什么?是那个栗师首创的还是 下面的cfy\_cjxwh给的论文里面的?但是他给的链接失...

单旋SPLAY的优化 qq\_19735179: 求给个QQ。。。 我想了解一下源代码。。看源代码更好理解一下。。我的QQ 329526532

在windows右键菜单中加入"新建 CPP"并创建模板 Ric\_shooter: 原来光注销不行,

还要重启...

在windows右键菜单中加入"新建 CPP"并创建模板 Ric\_shooter: 为什么我的不行啊

楼主!

光线追踪——最终版 loyave:问一下,你这个读入还有 输出的文件是什么...

光线追踪——最终版 loyave: 膜拜一下

了~\(≧▽≦)/~

【基于归并排序的分治】统计问

oy1997: 除了膜拜还能说什么.这 算法完全开辟了一种新的维护数 据结构题的思路

【快速傅里叶变换】FFT高精乘 Yoki飒: 盾哥程序不错哦......受教

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 | 合作伙伴 | 论坛反馈

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 |

江苏乐知网络技术有限公司 提供商务支持

京 ICP 证 09002463 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved

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 | 合作伙伴 | 论坛反馈

杂志客服 微博客服 网站客服 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | webmaster@csdn.net

江苏乐知网络技术有限公司 提供商务支持

京 ICP 证 09002463 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved

File failed to load: file:///C:/Users/wkl31415926/OneDrive/IO\_wkl\_/%E5%85%A5%E9%97%A8%E9%A2%98%E7%9B%AE%E6%95%B4%E7%90%86/%E5%87%A0%E4%BD%95/%