

其它

目录

1	快读	1
2	对拍	1
3	华容道	3
4	希尔伯特曲线	3
5	约瑟夫环	4
5.1	一般方法	4
5.2	函数图像解	5

1 快读

```
1 inline int read(){//如果是long long, 这行和下面一行int改ll
2     int x=0,f=1;
3     char ch=getchar();
4     while(ch<'0' || ch>'9'){
5         if(ch=='-')
6             f=-1;
7         ch=getchar();
8     }
9     while(ch>='0' && ch<='9'){
10         x=(x<<1)+(x<<3)+(ch^48);
11         ch=getchar();
12     }
13     return x*f;
14 }//打死我都不用!!!!
```

2 对拍

```
1 //-----data.cpp-----
2 int main()
3 {
4     freopen("in", "w", stdout);
5     srand(time(0));
6     int n,m,q;
7     n = rand()%100000;
8     m = rand()%100000;
```

```

9      q = rand()%100000;
10     printf("%d %d %d\n",n,m,q);
11     for(int i = 1;i <= q;i++){
12         int a = rand()%n+1;
13         int b = rand()%n+1;
14         int c = rand()%2;
15         printf("%d %d %d\n",a,b,c);
16     }
17     return 0;
18 }
19
20 //-----1.cpp&&2.cpp-----
21 int main()
22 {
23     freopen("in","r",stdin);
24     freopen("1.out","w",stdout);
25     //freopen("2.out","w",stdout);
26     .....
27 }
28
29 //-----duipai.cpp-----
30 int main()//Windows
31 {
32     int cases = 0;
33     do{
34         if(cases) printf("#%d AC\n",cases);
35         cases++;
36         system("data.exe > data.txt");
37         system("1.exe < data.txt > 1.txt");
38         system("2.exe < data.txt > 2.txt");
39     }while(!system("fc 1.txt 2.txt"));
40     printf("#%d WA",cases);
41     return 0;
42 }
43 int main()//Linux
44 {
45     int i;
46     for (i=1;i<=1000;i++)
47     {
48         system("./data");
49         system("./1");
50         system("./2");
51         printf("%d : ",i);
52         if (system("diff 1.out 2.out"))
53             {
54                 printf("WA\n");
55                 return 0;
56             }
57         else printf("AC\n");

```

```

58     }
59     return 0;
60 }

```

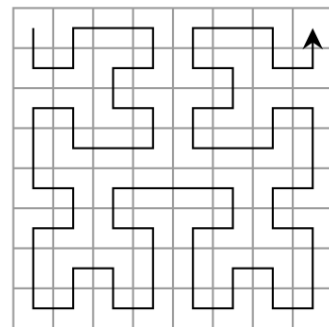
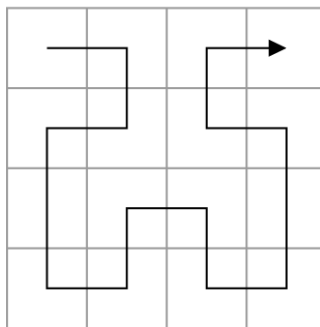
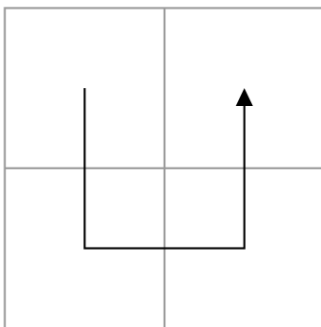
3 华容道

```

1 //判断是否有解
2 int map[16],ans=0;
3 for(int i=0;i<16;i++){
4     scanf("%d",&map[i]);
5     if(!map[i])
6         ans+=6-i%4-i/4;
7     for(int j=0;j<i;j++)
8         if(map[j]>map[i])
9             ans++;
10 }
11 if(ans&1)
12     printf("Yes\n");
13 else
14     printf("No\n");

```

4 希尔伯特曲线



```

1 #define ll long long
2 int two[50]; //2的次方
3 ll f(int n, int x, int y) { //返回第几位
4     if (n == 0) return 1;
5     int m = two[n-1]; //1 << (n - 1); //2的n-1次方
6     if (x <= m && y <= m) {
7         return f(n - 1, y, x);
8     }
9     if (x > m && y <= m) {
10        return 3LL * m * m + f(n - 1, m-y+ 1, m * 2 - x + 1); // 3LL表示ll 类型的3

```

```

11     }
12     if (x <= m && y > m) {
13         return 1LL * m * m + f(n - 1, x, y - m);
14     }
15     if (x > m && y > m) {
16         return 2LL * m * m + f(n - 1, x - m, y - m);
17     }
18 }
19 const int SIZE=1e6+50;
20 struct node{                                //用于存点
21     int x,y;
22     ll no;
23 }p[SIZE];
24 int main() {
25     int n,int k;
26     scanf("%d%d",&n,&k);
27     two[0]=1;                                //two[1]=2;
28     for(int i=1;i<=32;i++){
29         two[i]=2*two[i-1];
30     }
31     for(int i=1;i<=n;i++){
32         scanf("%d%d",&p[i].y,&p[i].x);        //注意y,x的读入顺序!
33         p[i].no=f(k,p[i].x,p[i].y);          //用于存点的编号
34     }
35 }

```

5 约瑟夫环

5.1 一般方法

```

1  /* *  n个 人(编号 1...n),先去掉第m个数,然后从m+1个开始报1, *
2  报到k的退出,剩下的 人继续从1开始报数.求胜利者的编号. */
3  int main(int argc, const char *argv[])
4  {
5      int n, k, m;
6      while (cin >> n >> k >> m, n || k || m)
7      {
8          int i, d, s = 0;
9          for (i = 2; i <= n; i++)
10             {
11                 s = (s + k) % i;
12             }
13             k = k % n;
14             if (k == 0)
15             {
16                 k = n;
17             }

```

```

18     d = (s + 1) + (m - k);
19     if (d >= 1 && d <= n)
20     {
21         cout << d << '\n';
22     }
23     else if (d < 1)
24     {
25         cout << n + d << '\n';
26     }
27     else if (d > n)
28     {
29         cout << d % n << '\n';
30     }
31 }
32 return 0;
33 }

```

5.2 函数图像解

```

1  /* * n 个 人数到 k 出列列，后剩下的人编号 */
2  unsigned long long n, k;
3  int main()
4  {
5      cin >> n >> k;
6      long long y = k % 2;
7      long long x = 2, t = 0;
8      long long z1 = y, z2 = x;
9      while (x <= n)
10     {
11         z1 = y;
12         z2 = x;
13         t = (x - y) / (k - 1);
14         if (t == 0)
15         {
16             t++;
17         }
18         y = y + t * k - ((y + t * k) / (x + t)) * (x + t);
19         x += t;
20     }
21     cout << (z1 + (n - z2) * k) % n + 1 << endl;
22     return 0;
23 }

```