Contents

1 基础/配置/黑科技

1.1 一般母版

```
/*
  Time:
  Prob:
  By RogerRo
 */
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#include<vector>
#include<queue>
#include<set>
#include<map>
#include<cmath>
#include<algorithm>
#include<ctime>
#include<bitset>
#define ll long long
#define tr(i,l,r) for((i)=(l);(i)<=(r);++i)
#define rtr(i,r,l) for((i)=(r);(i)>=(l);--i)
#define oo 0x7F7F7F7F
using namespace std;
int read()
{
    int x=0; bool f=0;
    char ch=getchar();
    while (ch<'0'||ch>'9') {f|=ch=='-'; ch=getchar();}
    while (ch>='0'&&ch<='9') {x=(x<<3)+(x<<1)+ch-'0'; ch=getchar();}
    return (x^-f)+f;
}
void write(int x)
{
    char a[20],s=0;
    if (x==0){putchar('0'); return ;}
    if (x<0) {putchar('-'); x=-x;}
    while (x) {a[s++]=x%10+'0'; x=x/10;}
    while (s--) putchar(a[s]);
}
void writeln(int x){write(x); putchar('\n');}
int main()
{

    return 0;
}
```

1.2 黑科技

```
//===================万能头文件===================
#include<bits/stdc++.h>
//===================强制O2优化===================
#pragma GCC optimize(2)
//===================开栈===================
//g++开栈  放在main开头
int __size__=256<<20;//256MB
char *__p__=(char*)malloc(__size__)+__size__;
__asm__ __volatile__("movq %0,%%rsp\n"::"r"(__p__));
//c++开栈
#pragma comment(linker,"/STACK:102400000,102400000")
//===================C++IO加速===================
#include <iomanip>
ios_base::sync_with_stdio(false);
//===================大数mulmod===================
//int128法
ll mulmod(__int128 x,__int128 y,__int128 mod)    //同理存在__float128
{
    return x*y%mod;
}

//快速乘法
ll mulmod(ll x,ll y,ll mod)
{
  ll ret = 0;
  for(;y;y>>=1)
  {
    if (y&1) ret=(ret+x)%mod;
    x=(x+x)%mod;
  }
  return ret;
}

//汇编法
ll mulmod(ll x,ll y,ll mod) //注意！必须保证x，y都比mod小；可long，不可int
{
    ll ans=0;
    __asm__
    (
        "movq %1,%%rax\n imulq %2\n idivq %3\n"
        :"=d"(ans):"m"(x),"m"(y),"m"(mod):"%rax"
    );
    return ans;
}
//===================其它小东西===================
int __gcd(int x,int y)  //<algorithm>且g++才能用
```

1.3 位运算

```
//===================枚举i的非空子集j===================
for(j=i;j;j=(j-1)&i);
//===================下一个1的个数相等的数===================
int snoob1(int x)
{
    int y=x&-x,z=x+y;
    return z|((x^z)>>2)/y;
```

```
8  }
9  int snoob2(int x)    //g++
10 {
11     int t=x|(x-1);
12     return (t+1)|(((~t&-~t)-1)>>(__builtin_ctz(x)+1));
13 }
14 //======================按位反转=======================
15 int reverse(int x)
16 {
17     x=((x&0x55555555)<<1)|((x&0xAAAAAAAA)>>1);
18     x=((x&0x33333333)<<2)|((x&0xCCCCCCCC)>>2);
19     x=((x&0x0F0F0F0F)<<4)|((x&0xF0F0F0F0)>>4);
20     x=((x&0x00FF00FF)<<8)|((x&0xFF00FF00)>>8);
21     x=((x&0x0000FFFF)<<16)|((x&0xFFFF0000)>>16);
22     return x;
23 }
24 //========注意！！以下g++下才能用；ll则在函数名后加ll========
25 int __builtin_popcount(unsigned int x); //1的个数
26 int __builtin_clz(unsigned int x);        //前缀0的个数
27 //x为int时，31-__builtin_clz(x) 等价于 int(log(x)/log(2))
28 int __builtin_ctz(unsigned int x);        //后缀0的个数
29 int __builtin_parity(unsigned int x);    //1的个数%2
```

1.4 离散化

```
1  //dc[1,2,...]=[x1,x2,..]; rdc(x1,x2,...)=1,2,...
2  int n,a[maxn],dc[maxn];
3  int rdc(int x){return lower_bound(dc+1,dc+num+1,x)-dc;}
4  void init()
5  {
6      //...
7      memcpy(dc,a,(n+1)*sizeof(int));
8      sort(dc+1,dc+n+1);
9      num=unique(dc+1,dc+n+1)-(dc+1);
10 }
```

1.5 Linux 对拍

```
1  g++ $2 -o 1.out
2  g++ $3 -o 2.out
3  cnt=0;
4  while true; do
5  g++ $1 -o dm.out
6  ./dm.out>dm.txt
7  ./1.out<dm.txt>1.txt
8  ./2.out<dm.txt>2.txt
9  if diff 1.txt 2.txt; then let "cnt+=1"; echo ${cnt};
10 else exit 0;
11 fi
12 done
```

1.6 vimrc

```
1  runtime! debian.vim
2
3  if has("syntax")
4    syntax on
5  endif
6
7  if filereadable("/etc/vim/vimrc.local")
8    source /etc/vim/vimrc.local
9  endif
10
11
12 colo torte
13 set nu
14 set ts=4
15 set sw=4
16 map <C-A> ggVG"+y
17 map <F2> :w<CR>
18 map <F3> :browse e<CR>
19 map <F4> :browse vsp<CR>
20 map <F5> :call Run()<CR>
21 func! Run()
22   exec "w"
23   exec "!g++ -Wall % -o %<"
24   exec "!./%<"
25 endfunc
```

1.7 gdb

```
1  //g++ -g a.cpp -o a;gdb --args a 1
2  int main(int gdb)
3  {
4    if (gdb>1)
5    {
6      freopen("in","r",stdin);
7      freopen("out","w",stdout);
8    }
9
10   //...
11
12   return 0;
13 }
```

2 数学

2.1 分数类

```
1  int gcd(int x,int y){return y?gcd(y,x%y):x;}
2  struct frac
3  {
```

```
 4    int x,y;   //分子x，分母y，符号放在x
 5    frac adjust()
 6    {
 7      if (!y) {x=0; return *this;}
 8      if (!x) {y=1; return *this;}
 9      int sg=(x>0?1:−1)*(y>0?1:−1);
10      int t=gcd(x=abs(x),y=abs(y)); x=x/t*sg; y/=t;
11      return *this;
12    }
13    frac(){}
14    frac(int a,int b){x=a;y=b;this−>adjust();}
15    frac(char *a,bool improp=0) //improp假分数
16    {
17        int t,sg=1;
18        if (*a=='−') {sg=−1; a++;}
19        if (improp&&strchr(a,' ')) {sscanf(a,"%d %d/%d",&t,&x,&y); x+=t*y;}
20        else if (strchr(a,'/')) sscanf(a,"%d/%d",&x,&y);
21        else {sscanf(a,"%d",&x); y=1;}
22        x*=sg; this−>adjust();
23    }
24    char* c_str(bool improp=0)
25    {
26      char *res=new char[50](),t[50];
27      this−>adjust();
28      if (x==0) {res[0]='0'; return res;}
29      if (x<0) {strcat(res,"−"); x=−x;}
30      if (improp&&x/y&&x%y){sprintf(t,"%d ",x/y); strcat(res,t); x%=y;}
31      sprintf(t,"%d",x); strcat(res,t);
32      if (y!=1) {sprintf(t,"/%d",y); strcat(res,t);}
33      return res;
34    }

36      bool friend operator==(frac a,frac b){a.adjust();b.adjust();return a.x==b.
          x&&a.y==b.y;}
37      bool friend operator!=(frac a,frac b){a.adjust();b.adjust();return !(a.x==
          b.x&&a.y==b.y);}
38      bool friend operator>(frac a,frac b){if(a.x*b.x<=0)return a.x>b.x;return a
          .x*b.y>b.x*a.y;}
39      bool friend operator<(frac a,frac b){if(a.x*b.x<=0)return a.x<b.x;return a
          .x*b.y<b.x*a.y;}
40    frac friend operator+(frac a,frac b){return frac(a.x*b.y+a.y*b.x,a.y*b.y).
        adjust();}
41    frac friend operator−(frac a,frac b){return frac(a.x*b.y−a.y*b.x,a.y*b.y).
        adjust();}
42    frac friend operator*(frac a,frac b){return frac(a.x*b.x,a.y*b.y).adjust();}
43    frac friend operator/(frac a,frac b){return frac(a.x*b.y,a.y*b.x).adjust();}
44  };
45  const frac nonfrac=frac(0,0);
46  const frac zerofrac=frac(0,1);
```

2.2 高精度类

```
 1  //要sqrt就一定要len和dcm是偶数
 2  //不可以出现如big x=y;的东西，必须分开成big x;x=y;
```

```
 3  #define len 3000
 4  #define dcm 3000
 5  void carry(int*x,int y){*(x−1)+=((*x+=y)+10000)/10−1000;*x=(*x+10000)%10;}
 6  struct big
 7  {
 8      int _[len+2];
 9
10      int& operator[](int x){return _[x];}
11      big(){memset(_,0,sizeof(int)*(len+2));}
12      big(char*x)
13      {
14          memset(_,0,sizeof(int)*(len+2));
15          char *y=x+strlen(x)−1,*z=strchr(x,'.'),*i;
16          if (!z) z=y+1;
17          int t=dcm−(z−x);
18          tr(i,x,y) if(i!=z&&t>=1&&t<=len) _[++t]=*i−'0';
19      }
20
21      big& operator=(const big&x){memcpy(_,x._,sizeof(int)*(len+2));return *this
          ;}
22      char* c_str()
23      {
24          char *s=new char[len]; int l,r,i=0,k;
25          tr(l,1,len) if(_[l]>0||l==dcm) break;
26          rtr(r,len,1) if(_[r]>0||r==dcm) break;
27          tr(k,l,r){if(k==dcm+1)s[i++]='.';s[i++]=_[k]+'0';}
28          s[i]=0; return s;
29      }
30
31      friend int comp(big x,big y) //0(len)
32      {
33          int i;
34          tr(i,1,len) if (x[i]!=y[i]) break;
35          return i>len?0:(x[i]>y[i]?1:−1);
36      }
37      friend big operator+(big x,big y) //0(len)
38      {
39          big z; int i;
40          rtr(i,len,1) carry(&z[i],x[i]+y[i]);
41          return z;
42      }
43      friend big operator−(big x,big y) //0(len)
44      {
45          big z; int i;
46          rtr(i,len,1) carry(&z[i],x[i]−y[i]);
47          return z;
48      }
49      friend big operator*(big x,big y) //0(len^2)
50      {
51          big z; int i,j;
52          rtr(i,len,1) rtr(j,min(dcm+len−i,len),max(dcm+1−i,1))
53              carry(&z[i+j−dcm],x[i]*y[j]);
54          return z;
55      }
56      friend big operator/(big x,big y) //0(len^2)
57      {
```

```
58      big z,t,tmp[10]; int i,j,k;
59      tr(k,1,9) tmp[k]=tmp[k-1]+y;
60      tr(j,1,len-dcm) t[j+dcm]=x[j];
61      j--;
62      tr(i,1,len)
63      {
64          tr(k,1,len-1) t[k]=t[k+1];
65          t[len]=++j<=len?x[j]:0;
66          tr(k,1,9) if (comp(tmp[k],t)>0) break;
67          z[i]=--k;
68          t=t-tmp[k];
69      }
70      return z;
71  }
72  friend int sqrt_deal(big&y,int a,int b,int l)
73  {
74      int t=a+y[b]%10-9;
75      if(2*b>l)t-=(y[2*b-l])/10;
76      if (b>=0&&!(a=sqrt_deal(y,t/10,b-1,l))) y[b]+=(t+999)%10-y[b]%10;
77      return a;
78  }
79  friend big sqrt(big x) //O(len^2)
80  {
81      int l,t=dcm/2; big y,z; y=x;
82      for(l=1;l<=len;l++)
83      {
84          y[++l]+=10;
85          while (!sqrt_deal(y,0,l,l)) y[l]+=20;
86          z[++t]=y[l]/20; y[l]-=10;
87      }
88      return z;
89  }
90  friend big floor(big x)
91  {
92      big z; z=x; int i;
93      tr(i,dcm+1,len) z[i]=0;
94      return z;
95  }
96  friend big ceil(big x){return comp(x,floor(x))==0?x:floor(x+big("1"));}
97 };
```

2.3 筛素数-欧拉筛法

$O(N)$

```
1  int prime[maxm],a[n];
2  bool pprime[n];
3  void EulerPrime()
4  {
5    int i,j;
6    tr(i,2,n) pprime[i]=1;
7    tr(i,2,n)
8    {
9      if (pprime[i]) prime[++m]=i;
10     tr(j,1,m)
```

```
11     {
12        if (i*prime[j]>n) break;
13        pprime[i*prime[j]]=0;
14        if (i%prime[j]==0) break;
15     }
16   }
17 }
```

2.4 线性方程组-高斯消元

$O(N^3)$

```
1  //待测
2  int n,m;
3  frac a[maxn][maxm],ans[maxn];
4  void gauss()
5  {
6      int i=1,j,k,x,y,arb=m;
7      frac t;
8      tr(j,1,m)
9      {
10         tr(k,i,n) if (a[k][j]!=zerofrac) break;
11         if (k>n) continue;
12         arb--;
13         if (k!=i) tr(y,j,m+1) swap(a[i][y],a[k][y]);
14         tr(x,1,n) if(x!=i)
15         {
16             t=a[x][j]/a[i][j];
17             tr(y,j,m+1) a[x][y]=a[x][y]-a[i][y]*t;
18         }
19         if (++i>n) break;
20     }
21     //—————————
22     tr(i,1,n)
23     {
24         tr(j,1,m) if (a[i][j]!=zerofrac) break;
25         if (j>m&&a[i][j]!=zerofrac) {printf("No Solution.\n"); return;}
26     }
27     if (arb) {printf("Arbitrary constants: %d\n",arb); return;}
28     //—————————
29     rtr(i,m,1)
30     {
31         ans[i]=a[i][m+1];
32         rtr(j,m,i+1) ans[i]=ans[i]-ans[j]*a[i][j];
33         ans[i]=ans[i]/a[i][i];
34     }
35     tr(i,1,m) printf("x[%d] = %s\n",i,ans[i].c_str());
36 }
```

2.5 高阶代数方程求根-求导

$O(N^3 * S)$，S 取决于精度

```
1  //求导至最高次为t时，a[t][i]表x^i的系数，ans[t]记录根；oo依题而定
2  double a[maxn][maxn],ans[maxn][maxn];
3  int n,anss[maxn];
4  double get(int x,double y)
5  {
6      int i; double res=0;
7      rtr(i,x,0) res=res*y+a[x][i];
8      return res;
9  }
10 void dich(int x,double ll,double rr)
11 {
12     if (cmp(get(x,ll))==0){ans[x][++anss[x]]=ll;return;}
13     if (cmp(get(x,rr))==0){ans[x][++anss[x]]=rr;return;}
14     if (cmp(get(x,ll)*get(x,rr))>0) return;
15     double l=ll,r=rr,mid;
16     while (l+eps<r) //亦可改为循环一定次数
17     {
18         int tl=cmp(get(x,l)),tm=cmp(get(x,mid=(l+r)/2));
19         if (tl==0) break;
20         if (tl*tm>=0) l=mid; else r=mid;
21     }
22     ans[x][++anss[x]]=l;
23 }
24 void work()
25 {
26     int i,j; double l,r;
27     rtr(i,n-1,1) tr(j,0,i) a[i][j]=a[i+1][j+1]*(j+1);
28     tr(i,0,n-1)
29     {
30         l=-oo;
31         tr(j,1,anss[i]){dich(i+1,l,r=ans[i][j]); l=r;}
32         dich(i+1,l,oo);
33     }
34     tr(i,1,anss[n]) printf("%.10lf\n",ans[n][i]);
35 }
```

3 几何

3.1 平面几何类包

下面提到皮克公式：$S = I + \frac{B}{2} - 1$ 描述顶点都在格点的多边形面积，I, B 分别为多边形内、边上格点

```
1  #define maxpn 10005
2  #define nonx 1E100
3  #define eps 1E-8
4  const double pi=acos(-1.0);
5  int cmp(double x)
6  {
7      if (x>eps) return 1;
8      if (x<-eps) return -1;
9      return 0;
10 }
11 double sqr(double a){return a*a;}
12 int gcd(int a,int b){return a%b==0?b:gcd(b,a%b);}
13 struct point
14 {
15     double x,y;
16     point(){}
17     point(double a,double b){x=a;y=b;}
18
19     friend point operator+(point a,point b){return point(a.x+b.x,a.y+b.y);}
20     friend point operator-(point a,point b){return point(a.x-b.x,a.y-b.y);}
21     friend point operator-(point a){return point(-a.x,-a.y);}
22     friend double operator*(point a,point b){return a.x*b.x+a.y*b.y;}
23     friend point operator*(double a,point b){return point(a*b.x,a*b.y);}
24     friend point operator*(point a,double b){return point(a.x*b,a.y*b);}
25     friend point operator/(point a,double b){return point(a.x/b,a.y/b);}
26     friend double operator^(point a,point b){return a.x*b.y-a.y*b.x;}
27     friend bool operator==(point a,point b){return cmp(a.x-b.x)==0&&cmp(a.y-b.y)==0;}
28
29     friend double sqr(point a){return a*a;}
30     friend double len(point a){return sqrt(sqr(a));}    //模长
31     friend point rotate(point a,double b){return point(a.x*cos(b)-a.y*sin(b),a.x*sin(b)+a.y*cos(b));}    //逆时针旋转
32     friend double angle(point a,point b){return acos(a*b/len(a)/len(b));}    //夹角
33     friend point reflect(point a,point b){return 2*a-b;}    //以a为中心对称
34 } ;
35 const point nonp=point(nonx,nonx);
36 point quad(double A,double B,double C)
37 {
38     double delta=sqr(B)-4*A*C;
39     if (delta<0) return nonp;
40     return point((-B-sqrt(delta))/(2*A),(-B+sqrt(delta))/(2*A));
41 }
42 struct line
43 {
44     point a,b;
45     line(){}
46     line(point pa,point pb){a=pa;b=pb;}
47     point dir(){return b-a;}
48
49     friend point proj(point a,line b){double t=(a-b.a)*b.dir()/sqr(b.dir());return point(b.a+t*b.dir());}    //垂足
50     friend double dist(point a,line b){return ((a-b.a)^(b.b-b.a))/len(b.dir());}    //点到线距离
51     friend bool onray(point a,line b){return cmp((a-b.a)^b.dir())==0&&cmp((a-b.a)*b.dir())>=0;} //判断点在射线上
52     friend bool onseg(point a,line b){return cmp((a-b.a)^b.dir())==0&&cmp((a-b.a)*(a-b.b))<=0;} //判断点在线段上
53     friend bool online(point a,line b){return cmp((a-b.a)^b.dir())==0;} //判断点在直线上
54     friend bool parallel(line a,line b){return cmp(a.dir()^b.dir())==0;}    //判断两线平行
55     friend point cross(line a,line b)    //线交
56     {
57         double t;
```

```
 58          if (cmp(t=a.dir()^b.dir())==0) return nonp;
 59          return a.a+((b.a−a.a)^b.dir())/t*a.dir();
 60      }
 61  } ;
 62  const line nonl=line(nonp,nonp);
 63  struct circle
 64  {
 65      point o; double r;
 66      circle(){}
 67      circle(point a,double b){o=a;r=b;}
 68
 69      friend double S(circle a){return pi*sqr(a.r);}   //面积
 70      friend double C(circle a){return 2*pi*a.r;} //周长
 71      friend line cross(line a,circle b)   //线圆交
 72    {
 73      point t=quad(sqr(a.dir()),2*a.dir()*(a.a−b.o),sqr(a.a−b.o)−sqr(b.r));
 74      if (t==nonp) return nonl;
 75      return line(a.a+t.x*a.dir(),a.a+t.y*a.dir());
 76    }
 77      friend int in(point a,circle b){double t=len(a−b.o);return t==b.r?2:t<b.r
            ;} //点与圆位置关系 0外 1内 2上
 78      //friend line cross(circle a,circle b){}
 79      //friend line tangent(point a,circle b){}
 80      //friend pair<line,line> tangent(circle a,circle b){}
 81      //friend double unionS(int n,circle*a) //圆面积并
 82      //{}
 83  } ;
 84  struct triangle//t 因triangle亦属polygon，故省去许多函数
 85  {
 86      point a,b,c;
 87      triangle(){}
 88      triangle(point ta,point tb,point tc){a=ta;b=tb;c=tc;}
 89
 90      friend double S(triangle a){return abs((a.b−a.a)^(a.c−a.a))/2;} //面积
 91      friend double C(triangle a){return len(a.a−a.b)+len(a.a−a.c)+len(a.a−a.c)
            ;} //周长
 92      friend circle outcircle(triangle a) //外接圆
 93      {
 94          circle res; point t1=a.b−a.a,t2=a.c−a.a;
 95          double t=2*t1^t2;
 96          res.o.x=a.a.x+(sqr(t1)*t2.y−sqr(t2)*t1.y)/t;
 97          res.o.y=a.a.y+(sqr(t2)*t1.x−sqr(t1)*t2.x)/t;
 98          res.r=len(res.o−a.a);
 99          return res;
100      }
101      friend circle incircle(triangle a)   //内切圆
102      {   circle res; double x=len(a.b−a.c),y=len(a.c−a.a),z=len(a.a−a.b);
103          res.o=(a.a*x+a.b*y+a.c*z)/(x+y+z);
104          res.r=dist(res.o,line(a.a,a.b));
105          return res;
106      }
107      friend point gc(triangle a){return (a.a+a.b+a.c)/3;}    //重心
108      friend point hc(triangle a){return 3*gc(a)−2*outcircle(a).o;}    //垂心
109  } ;
110  struct polygon
111  {
112      int n; point a[maxpn];  //逆时针！
113      polygon(){}
114      polygon(triangle t){n=3;a[1]=t.a;a[2]=t.b;a[3]=t.c;}
115      point& operator[](int _){return a[_];}
116
117      friend double S(polygon a)  //面积 O(n)
118      {
119          int i; double res=0;
120          a[a.n+1]=a[1];
121          tr(i,1,a.n) res+=a[i]^a[i+1];
122          return res/2;
123      }
124      friend double C(polygon a) //周长 O(n)
125      {
126          int i; double res=0;
127          a[a.n+1]=a[1];
128          tr(i,1,a.n) res+=len(a[i+1]−a[i]);
129          return res;
130      }
131      friend int in(point a,polygon b)  //点与多边形位置关系 0外 1内 2上 O(n)
132      {
133          int s=0,i,d1,d2,k;
134          b[b.n+1]=b[1];
135          tr(i,1,b.n)
136          {
137              if (onseg(a,line(b[i],b[i+1]))) return 2;
138              k=cmp((b[i+1]−b[i])^(b[i]−a));
139              d1=cmp(b[i].y−a.y);
140              d2=cmp(b[i+1].y−a.y);
141              s=s+(k>0&&d2<=0&&d1>0)−(k<0&&d1<=0&&d2>0);
142          }
143          return s!=0;
144      }
145      friend point gc(polygon a)  //重心 O(n)
146      {
147          double s=S(a); point t(0,0); int i;
148          if (cmp(s)==0) return nonp;
149          a[a.n+1]=a[1];
150          tr(i,1,a.n) t=t+(a[i]+a[i+1])*(a[i]^a[i+1]);
151          return t/s/6;
152      }
153      friend int pick_on(polygon a)   //皮克求边上格点数 O(n)
154      {
155          int s=0,i;
156          a[a.n+1]=a[1];
157          tr(i,1,a.n) s+=gcd(abs(int(a[i+1].x−a[i].x)),abs(int(a[i+1].y−a[i].y))
               );
158          return s;
159      }
160      friend int pick_in(polygon a){return int(S(a))+1−pick_on(a)/2;} //皮克求多
          边形内格点数 O(n)
161
162      //friend line convex_maxdist(polygon a){}
163      //friend line mindist(polygon a){} //a只是点集
164      //friend polygon convex_hull(polygon a){} //a只是点集 O(nlogn)
```

```
166    //friend int convex_in(point a,polygon b){} //0外 1内 2上 O(logn)
167    //friend polygon cross(polygon a,polygon b){}
168    //friend polygon cross(line a,polygon b){}
169    //friend double unionS(circle a,polygon b){}
170    friend circle mincovercircle(polygon a) //最小圆覆盖 O(n)
171    {
172        circle t; int i,j,k;
173        srand(time(0));
174        random_shuffle(a.a+1,a.a+a.n+1);
175        for(i=2,t=circle(a[1],0);i<=a.n;i++) if (!in(a[i],t))
176            for(j=1,t=circle(a[i],0);j<i;j++) if (!in(a[j],t))
177                for(k=1,t=circle((a[i]+a[j])/2,len(a[i]−a[j])/2);k<j;k++) if
                        (!in(a[k],t))
178                    t=outcircle(triangle(a[i],a[j],a[k]));
179        return t;
180    }
181 } ;
```

4 DP

5 串

5.1 最长回文子串-Manacher

$O(N)$

```
1  //st，s都从1开始！
2  //     1 2 3 4 5 6 7 8
3  // st:  a b a
4  // s:   0 0 a 0 b 0 a 0
5  // a:   0 0 1 2 3 2 1 0
6  int a[2*maxl];
7  char st[maxl],s[2*maxl];
8  int manacher()
9  {
10     int l=strlen(st+1),i,Mm,Mr=0,ans=0;
11     memset(a,0,sizeof(a)); s[1]=0xFF;
12     tr(i,2,2*l+2) s[i]=(i&1)*st[i/2];
13     tr(i,1,2*l+2)
14     {
15         if (i<Mr) a[i]=min(a[2*Mm−i],Mr−i);
16         while (s[i−a[i]−1]==s[i+a[i]+1]) a[i]++;
17         if (i+a[i]>Mr) {Mr=i+a[i]; Mm=i;}
18         ans=max(ans,a[i]);
19     }
20     return ans;
21  }
22  int main()
23  {
24     gets(st+1); printf("%d\n",manacher());
25     return 0;
26  }
```

5.2 多模匹配-AC 自动机

求 n 个模式串中有多少个出现过，模式串相同算作多个，$O(\sum P_i + T)$

```
1  //maxt=文本串长，maxp=模式串长，maxn=模式串数
2  struct ac{int s,to[26],fail;} a[maxn*maxp];
3  int m,n;
4  char ts[maxp],s[maxt];
5  queue<int> b;
6  void clear(int x)
7  {
8      a[x].s=a[x].fail=0;
9      memset(a[x].to,0,sizeof(a[x].to));
10 }
11 void ins(char *st)
12 {
13     int i,x=0,c,l=strlen(st);
14     tr(i,0,l−1)
15     {
16         if (!a[x].to[c=st[i]−'a']) {a[x].to[c]=++m; clear(m);}
17         x=a[x].to[c];
18     }
19     a[x].s++;
20 }
21 void build()
22 {
23     int i,h,t;
24     tr(i,0,25) if (t=a[0].to[i]) b.push(t);
25     while (b.size())
26     {
27         h=b.front(); b.pop();
28         tr(i,0,25)
29         if (t=a[h].to[i])
30         {
31             a[t].fail=a[a[h].fail].to[i];
32             b.push(t);
33         } else a[h].to[i]=a[a[h].fail].to[i];
34     }
35 }
36 int cnt(char *st)
37 {
38     int i,x=0,c,t,cnt=0,l=strlen(st);
39     tr(i,0,l−1)
40     {
41         c=st[i]−'a';
42         while (!a[x].to[c]&&x) x=a[x].fail;
43         x=a[x].to[c];
44         for(t=x;t&&a[t].s>−1;t=a[t].fail) {cnt+=a[t].s; a[t].s=−1;}
45     }
46     return cnt;
47 }
48 void work()
49 {
50     int i;
51     m=0; clear(0);
52     scanf("%d",&n);
```

```
53      tr(i,1,n)
54      {
55          scanf("%s",ts); ins(ts);
56      }
57      build();
58      scanf("%s",s); printf("%d\n",cnt(s));
59  }
```

6 图/树

6.1 单源最短路-Dijkstra

不加堆，$O(V^2 + E)$

```
1   struct edge{int pre,x,y,d;} a[maxm];
2   int n,m,ah[maxn],d[maxn];
3   bool p[maxn];
4   void update(int x)
5   {
6     int e;
7     p[x]=true;
8     for(e=ah[x];e>-1;e=a[e].pre)
9       if (!p[a[e].y]&&(!d[a[e].y]||a[e].d+d[x]<d[a[e].y]))
10        d[a[e].y]=a[e].d+d[x];
11  }
12  void dijkstra()
13  {
14    int i,j,t;
15      memset(p,0,sizeof(p));
16    update(1);
17    d[0]=oo;
18    tr(i,2,n)
19    {
20      t=0;
21      tr(j,1,n) if (!p[j]&&d[j]&&d[j]<d[t]) t=j;
22      update(t);
23    }
24    printf("%d\n",d[n]);
25  }
```

加堆，$O(ElogE + V)$

```
1   typedef pair<int,int> pa;
2   struct edge{int pre,x,y,d;} a[maxm];
3   int n,m,ah[maxn],ans[maxn];
4   priority_queue<pa,vector<pa>,greater<pa> >d;
5   bool p[maxn];
6   void dijkstra()
7   {
8     int v,s,e;
9       memset(p,0,sizeof(p));
10    d.push(make_pair(0,1));
11    while(!d.empty())
12    {
```

```
13      v=d.top().second;
14      s=d.top().first;
15      d.pop();
16      if (p[v]) continue;
17      p[v]=1;
18      ans[v]=s;
19      for(e=ah[v];e>-1;e=a[e].pre)
20        if (!p[a[e].y]) d.push(make_pair(s+a[e].d,a[e].y));
21    }
22    printf("%d\n",ans[n]);
23  }
```

6.2 最短路-Floyd

$O(V^3 + E)$

```
1   void floyd()
2   {
3     int i,j,k;
4     tr(k,1,n) tr(i,1,n)
5       if (a[i][k]) tr(j,1,n)
6         if (i!=j&&a[k][j]&&(!a[i][j]||(a[i][j]&&a[i][k]+a[k][j]<a[i][j])))
7               a[i][j]=a[i][k]+a[k][j];
8   }
```

6.3 单源最短路-SPFA

不加优化，$O(VE + V^2) = O(kE)$

```
1   struct edge{int pre,x,y,d;} a[maxm];
2   int n,m,last[maxn],d[maxn],b[maxn];
3   bool p[maxn];
4   void spfa()
5   {
6     int h,t,e;
7     memset(d,0x7F,sizeof(d));
8       memset(p,0,sizeof(p));
9     b[0]=1; p[1]=1; d[1]=0;
10    h=n-1; t=0;
11    while (h!=t)
12    {
13      h=(h+1)%n;
14      for (e=last[b[h]];e>-1;e=a[e].pre)
15        if (d[a[e].x]+a[e].d<d[a[e].y])
16        {
17          d[a[e].y]=d[a[e].x]+a[e].d;
18          if (!p[a[e].y])
19          {
20            t=(t+1)%n;
21            b[t]=a[e].y;
22            p[a[e].y]=1;
23          }
24        }
25      p[b[h]]=0;
```

```
26      }
27    printf("%d\n",d[n]);
28  }
```

SLF+LLL 优化，$O(VE + V^2) = O(kE)$

```
1   //a从1开始！
2   struct edge{int pre,x,y,d;} a[maxm];
3   int n,m,last[maxn],d[maxn],b[maxn];
4   bool p[maxn];
5   void spfa()
6   {
7     int e,h,t,sum,num;
8     memset(d,0x7F,sizeof(d));
9       memset(p,0,sizeof(p));
10    b[0]=1; p[1]=1; d[1]=0;
11    sum=0; num=1;
12    h=0; t=0;
13    while (num)
14    {
15      while (d[h]*num>sum)
16      {
17        t=(t+1)%n;
18        b[t]=b[h];
19        h=(h+1)%n;
20      }
21      e=last[b[h]];
22      p[b[h]]=0;
23      num--;
24      sum-=d[a[e].x];
25      h=(h+1)%n;
26      for (;a[e].x;e=a[e].pre)
27        if (d[a[e].x]+a[e].d<d[a[e].y])
28        {
29          if (p[a[e].y]) sum-=d[a[e].y];
30          d[a[e].y]=d[a[e].x]+a[e].d;
31          sum+=d[a[e].y];
32          if (!p[a[e].y])
33          {
34            if (num && d[a[e].y]<d[b[h]])
35            {
36              h=(h+n-1)%n;
37              b[h]=a[e].y;
38            } else
39            {
40              t=(t+1)%n;
41              b[t]=a[e].y;
42            }
43            p[a[e].y]=1;
44            num++;
45          }
46        }
47    }
48    printf("%d\n",d[n]);
49  }
```

6.4 二分图最大匹配-匈牙利

$O(VE)$

```
1   struct edge{int x,y,pre;} a[maxm];
2   int nx,ny,m,last[maxn],my[maxn];
3   bool p[maxn];
4   int dfs(int x)
5   {
6     for (int e=last[x];e>-1;e=a[e].pre)
7       if (!p[a[e].y])
8       {
9         int y=a[e].y;
10        p[y]=1;
11        if (!my[y]||dfs(my[y])) return my[y]=x;
12      }
13    return 0;
14  }
15  void hungary()
16  {
17    int i,ans=0;
18    memset(my,0,sizeof(my));
19    tr(i,1,nx)
20    {
21      memset(p,0,sizeof(p));
22      if (dfs(i)) ans++;
23    }
24    printf("%d\n",ans);
25  }
```

6.5 有向图极大强连通分量-Tarjan 强连通

$O(V + E)$

```
1   //ds，ss，gs分别是dfn，sta，group计数器；group记所属分量号码，size记分量大小；
        insta记是否在栈中
2   struct edge{int x,y,pre;} a[maxm];
3   int n,m,ah[maxn],ds,dfn[maxn],low[maxn],ss,sta[maxn],gs,group[maxn],size[maxn
        ];
4   bool insta[maxn];
5   void tarjan(int x)
6   {
7     int e,y,t;
8     dfn[x]=low[x]=++ds;
9     sta[++ss]=x; insta[x]=1;
10    for(e=ah[x];e>-1;e=a[e].pre)
11    {
12      if (!dfn[y=a[e].y]) tarjan(y);
13      if (insta[y]) low[x]=min(low[x],low[y]);
14    }
15    if (low[x]==dfn[x])
16      for(gs++,t=0;t!=x;t=sta[ss--]) {group[sta[ss]]=gs; size[gs]++;}
17  }
18  void work()
19  {
```

```
20    ds=ss=gs=0;
21    int i; tr(i,1,n) if (!dfn[i]) tarjan(i);
22 }
```

6.6 最大流-iSAP

简版（无 BFS，递归，gap，cur），$O(V^2 * E)$

```
1  struct edge{int x,y,c,f,pre;} a[2*maxm];
2  int n,mm,m,last[maxn],d[maxn],gap[maxn],cur[maxn],ans;
3  void newedge(int x,int y,int c,int f)
4  {
5    m++;
6    a[m].x=x; a[m].y=y; a[m].c=c; a[m].f=f;
7    a[m].pre=last[x]; last[x]=m;
8  }
9  void init()
10 {
11   int i,x,y,c;
12   m=-1;
13   memset(last,-1,sizeof(last));
14   tr(i,1,mm)
15   {
16     x=read(); y=read(); c=read();
17     newedge(x,y,c,0);
18     newedge(y,x,c,c);
19   }
20   tr(i,1,n) cur[i]=last[i];
21     memset(d,0,sizeof(d));
22   memset(gap,0,sizeof(gap));
23   gap[0]=n;
24   ans=0;
25 }
26 int sap(int x,int flow)
27 {
28   int e,t;
29   if (x==n) return flow;
30   for (e=cur[x];e!=-1;e=a[e].pre)
31     if (a[e].f<a[e].c && d[a[e].y]+1==d[x])
32     {
33       cur[x]=e;
34       if (t=sap(a[e].y,min(flow,a[e].c-a[e].f)))
35       {
36         a[e].f+=t; a[e^1].f-=t; return t;
37       }
38     }
39   if (--gap[d[x]]==0) d[n]=n;
40   d[x]=n;
41   for (e=last[x];e!=-1;e=a[e].pre)
42     if (a[e].f<a[e].c) d[x]=min(d[x],d[a[e].y]+1);
43   cur[x]=last[x];
44   ++gap[d[x]];
45   return 0;
46 }
47 int work()
```

```
48 {
49   while (d[n]<n) ans+=sap(1,oo);
50 }
```

完全版（有 BFS，非递归，gap，cur），$O(V^2 * E)$

```
1  int n,mm,m,ans,last[maxn],cur[maxn],pre[maxn],d[maxn],gap[maxn],b[maxn];
2  bool p[maxn];
3  struct edge{int x,y,c,f,pre;} a[2*maxm];
4  void newedge(int x,int y,int c,int f)
5  {
6    m++;
7    a[m].x=x; a[m].y=y; a[m].c=c; a[m].f=f;
8    a[m].pre=last[x]; last[x]=m;
9  }
10 void init()
11 {
12   int i,x,y,c;
13   m=-1;
14   memset(last,-1,sizeof(last));
15   tr(i,1,mm)
16   {
17     x=read(); y=read(); c=read();
18     newedge(x,y,c,0);
19     newedge(y,x,c,c);
20   }
21 }
22 int aug()
23 {
24   int x,flow=a[cur[1]].c-a[cur[1]].f;
25   for (x=pre[n];x>1;x=pre[x]) flow=min(flow,a[cur[x]].c-a[cur[x]].f);
26   return flow;
27 }
28 void bfs()
29 {
30   int h,t,e;
31   memset(p,0,sizeof(p));
32   b[1]=n; p[n]=1;
33   h=0; t=1;
34   while (h<t)
35   {
36     h++;
37     for (e=last[b[h]];e!=-1;e=a[e].pre)
38       if (a[e].c==a[e].f && !p[a[e].y])
39       {
40         b[++t]=a[e].y;
41         p[a[e].y]=1;
42         d[a[e].y]=d[a[e].x]+1;
43       }
44   }
45 }
46 void sap()
47 {
48   int x,e,flow;
49   memset(d,0,sizeof(d));
50   memset(gap,0,sizeof(gap));
```

```
51    bfs();
52    tr(x,1,n) gap[d[x]]++;
53    ans=0;
54    tr(x,1,n) cur[x]=last[x];
55    x=1; pre[1]=1;
56    while (d[1]<n)
57    {
58      for (e=cur[x];e!=−1;e=a[e].pre)
59        if (d[x]==d[a[e].y]+1 && a[e].f<a[e].c)
60        {
61          cur[x]=e;
62          pre[a[e].y]=x;
63          x=a[e].y;
64          break;
65        }
66      if (e==−1)
67      {
68        if (!(−−gap[d[x]])) return;
69        cur[x]=last[x];
70        d[x]=n;
71        for (e=last[x];e!=−1;e=a[e].pre)
72          if (a[e].f<a[e].c) d[x]=min(d[x],d[a[e].y]+1);
73        gap[d[x]]++;
74        x=pre[x];
75      }
76      if (x==n){
77        flow=aug();
78        for (x=pre[x];x>1;x=pre[x])
79        {
80          a[cur[x]].f+=flow; a[cur[x]^1].f−=flow;
81        }
82        a[cur[x]].f+=flow; a[cur[x]^1].f−=flow;
83        ans+=flow;
84        x=1;
85      }
86    }
87 }
```

6.7   最小生成树-Prim

不加堆，$O(V + E)$

```
1  struct edge{int x,y,d,pre;} a[maxm];
2  int n,m,ah[maxn],d[maxn];
3  bool p[maxn];
4  void prim()
5  {
6    int i,j,x,y,e,ans=0;
7    memset(d,0x7f,sizeof(d)); d[1]=0;
8    memset(p,0,sizeof(p));
9    tr(i,1,n)
10   {
11     x=0;
12     tr(j,1,n) if (!p[j]&&d[j]<d[x]) x=j;
13     ans+=d[x];
```

```
14     p[x]=1;
15     for(e=ah[x];e>−1;e=a[e].pre)
16       if (!p[y=a[e].y]) d[y]=min(d[y],a[e].d);
17   }
18   printf("%d\n",ans);
19 }
```

加堆，$O(V + E)$

```
1  struct edge{int x,y,d,pre;} a[maxm];
2  typedef pair<int,int> pa;
3  priority_queue<pa,vector<pa>,greater<pa> >d;
4  int n,m,ah[maxn];
5  bool p[maxn];
6  void prim()
7  {
8    int i,x,y,e,ans=0;
9    pa t;
10   while (!d.empty()) d.pop();
11   d.push(make_pair(0,1));
12   memset(p,0,sizeof(p));
13   tr(i,1,n)
14   {
15         while (!d.empty()&&p[d.top().second]) d.pop();
16     t=d.top();
17     ans+=t.first;
18     p[x=t.second]=1;
19     for(e=ah[x];e>−1;e=a[e].pre)
20       if (!p[y=a[e].y]) d.push(make_pair(a[e].d,y));
21   }
22   printf("%d\n",ans);
23 }
```

6.8   最小生成树-Kruskal

$O(ElogE + E)$

```
1  //a从1开始！
2  struct edge{int x,y,d;} a[maxm];
3  bool cmp(edge a,edge b){return a.d<b.d;}
4  int n,i,j,m,fa[maxn];
5  int gfa(int x){return x==fa[x]?x:fa[x]=gfa(fa[x]);}
6  void kruskal()
7  {
8    int ans,fx,fy;
9    sort(a+1,a+m+1,cmp);
10   tr(i,1,n) fa[i]=i;
11   ans=0;
12   tr(i,1,m)
13     if ((fx=gfa(a[i].x))!=(fy=gfa(a[i].y)))
14     {
15       fa[fx]=fy;
16       ans+=a[i].d;
17     }
18   printf("%d\n",ans);
```

```
19 | }
```

6.9 树的直径-BFS

$O(N)$

```
1  | struct edge{int x,y,d,pre;} a[2*maxn];
2  | int n,m,ah[maxn],d0[maxn],d1[maxn],b[maxn];
3  | bool p[maxn];
4  | void bfs(int root,int *d)
5  | {
6  |   int h,t,e,y;
7  |   memset(p,0,sizeof(p));
8  |   h=0; t=1;
9  |   b[1]=root;
10 |   p[root]=1;
11 |   while (h<t)
12 |   {
13 |     h++;
14 |     for (e=ah[b[h]];e>-1;e=a[e].pre)
15 |       if (!p[y=a[e].y])
16 |       {
17 |         b[++t]=y;
18 |         p[y]=1;
19 |         d[y]=d[a[e].x]+a[x].d;
20 |       }
21 |   }
22 | }
23 | void work()
24 | {
25 |   int i,s1,s2;
26 |     memset(d0,0,sizeof(d0));
27 |   memset(d1,0,sizeof(d1));
28 |   bfs(1,d0); s1=1; tr(i,1,n) if (d0[i]>d0[s1]) s1=i;
29 |   bfs(s1,d1); s2=1; tr(i,1,n) if (d1[i]>d1[s2]) s2=i;
30 |     printf("%d %d %d\n",s1,s2,d1[s2]);
31 | }
```

6.10 LCA-TarjanLCA

$O(N + Q)$

```
1  | struct query{int x,y,pre,lca;} b[2*maxq];
2  | struct edge{int x,y,pre,d;} a[2*maxn];
3  | int n,q,am,bm,ah[maxn],bh[maxn],fa[maxn],dep[maxn];
4  | bool p[maxn];
5  | int gfa(int x){return fa[x]==x?x:fa[x]=gfa(fa[x]);}
6  | void tarjan(int x,int depth)
7  | {
8  |     int tmp,y;
9  |     p[x]=1;
10 |     dep[x]=depth;
11 |     for(tmp=ah[x];tmp>-1;tmp=a[tmp].pre)
12 |         if (!p[y=a[tmp].y])
```

```
13 |         {
14 |             tarjan(y,depth+a[tmp].d);
15 |             fa[y]=x;
16 |         }
17 |     for(tmp=bh[x];tmp>-1;tmp=b[tmp].pre)
18 |         if (p[y=b[tmp].y]) b[tmp].lca=b[tmp^1].lca=gfa(y);
19 | }
20 | void work()
21 | {
22 |     memset(dep,0,sizeof(dep));
23 |     memset(p,0,sizeof(p));
24 |     tarjan(1,0);
25 |     int i; tr(i,0,q-1) writeln(dep[b[2*i].x]+dep[b[2*i].y]-2*dep[b[2*i].lca]);
26 | }
```

7 数据结构

7.1 并查集

```
1 | int gfa(int x){return(fa[x]==x?x:fa[x]=gfa(fa[x]));}
```

7.2 区间和 __ 单点修改区间查询-树状数组

$O(NlogN + QlogN)$

```
1  | int n,a[maxn],f[maxn];
2  | char tc;
3  | void modify(int x,int y)
4  | {
5  |     while (x<=n) {f[x]+=y; x+=x&-x;}
6  | }
7  | int sum(int x)
8  | {
9  |     int res=0;
10 |     while (x) {res+=f[x]; x-=x&-x;}
11 |     return res;
12 | }
13 | void work()
14 | {
15 |     int q,i,tx,ty;
16 |     n=read(); q=read();
17 |     memset(f,0,sizeof(f));
18 |     tr(i,1,n) modify(i,a[i]=read());
19 |     tr(i,1,q)
20 |     {
21 |         tc=getchar(); tx=read(); ty=read();
22 |         if (tc=='M') {modify(tx,ty-a[tx]); a[tx]=ty;}
23 |         else writeln(sum(ty)-sum(tx-1));
24 |     }
25 | }
```

7.3 区间和 _ 区间修改单点查询-树状数组

$O(NlogN + QlogN)$

```
1  int n,i,f[maxn];
2  void modify(int x,int y)
3  {
4      while (x) {f[x]+=y; x-=x&-x;}
5  }
6  int sum(int x)
7  {
8      int res=0;
9      while (x<=n) {res+=f[x]; x+=x&-x;}
10     return res;
11 }
12 void work()
13 {
14     int q,i;
15     n=read(); q=read();
16     memset(f,0,sizeof(f));
17     tr(i,1,q)
18     {
19         tc=getchar();
20         if (tc=='M') {modify(read()-1,-1); modify(read(),1);}
21         else writeln(sum(read()));
22     }
23 }
```

7.4 区间和-线段树

$O(NlogN + QlogN)$

```
1  struct node{int s,tag;} a[4*maxn];
2  int n;
3  void update(int t,int l,int r)
4  {
5      if (l!=r)
6      {
7          a[t<<1].tag+=a[t].tag;
8          a[t<<1|1].tag+=a[t].tag;
9      }
10     a[t].s+=(int)(r-l+1)*a[t].tag;
11     a[t].tag=0;
12 }
13 void add(int t,int l,int r,int x,int y,int z)
14 {
15     if (x<=l&&r<=y) {a[t].tag+=z; return ;}
16     a[t].s+=(int)(min(r,y)-max(l,x)+1)*z;
17     update(t,l,r);
18     int mid=(l+r)>>1;
19     if (x<=mid) add(t<<1,l,mid,x,y,z);
20     if (y>mid) add(t<<1|1,mid+1,r,x,y,z);
21 }
22 int sum(int t,int l,int r,int x,int y)
23 {
```

```
24     int res=0;
25     update(t,l,r);
26     if (x<=l&&r<=y) return a[t].s;
27     int mid=(l+r)>>1;
28     if (x<=mid) res+=sum(t<<1,l,mid,x,y);
29     if (y>mid) res+=sum(t<<1|1,mid+1,r,x,y);
30     return res;
31 }
32 void work()
33 {
34     int q,i,tx,ty; char tc;
35     n=read(); q=read();
36     tr(i,1,n) add(1,1,n,i,i,read());
37     tr(i,1,q)
38     {
39         tc=getchar(); tx=read(); ty=read();
40         if (tc=='A') add(1,1,n,tx,ty,read());
41         else writeln(sum(1,1,n,tx,ty));
42     }
43 }
```

7.5 平衡树-Treap

```
1  struct node
2  {
3      node* ch[2];
4      int x,y,size;
5      int chsize(int d){return ch[d]?ch[d]->size:0;}
6  } ;
7  node *root;
8  void newnode(node *&t,int x)
9  {
10     t=new node;
11     t->ch[0]=t->ch[1]=0;
12     t->x=x; t->y=rand(); t->size=1;
13 }
14 void rot(node *&t,int d)
15 {
16     node *tt=t->ch[!d];
17     t->ch[!d]=tt->ch[d];
18     tt->ch[d]=t;
19     tt->size=t->size;
20     t->size=t->chsize(0)+t->chsize(1)+1;
21     t=tt;
22 }
23 void ins(node *&t,int x)
24 {
25     if (!t) newnode(t,x);
26     else {
27         int d=t->x<x; ins(t->ch[d],x); ++t->size;
28         if (t->ch[d]->y<t->y) rot(t,!d);
29     }
30 }
31 void del(node *&t,int x)
```

```
32  {
33      if (x==t→x)
34      {
35          if (!t→ch[0]||!t→ch[1])
36          {
37              node *tt=t; t=t→ch[t→ch[0]==0]; delete(tt);
38              return;
39          } else
40          {
41              int d=t→ch[0]→y<t→ch[1]→y;
42              rot(t,d); del(t→ch[d],x);
43          }
44      } else del(t→ch[t→x<x],x);
45      --t→size;
46  }
47  node* kth(node *&t,int k)
48  {
49      if (k<=t→chsize(0)) return kth(t→ch[0],k);
50      else if(k>t→chsize(0)+1) return kth(t→ch[1],k-(t→chsize(0)+1));
51      else return t;
52  }
53  void work()
54  {
55      srand(time(0)); newnode(root,oo);
56      //...
57  }
```

7.6 区间第 k 大 __ 无修改-主席树

$O(NlogN + QlogN)$

```
1   struct node{int l,r,size;} a[maxm];
2   int n,q,m,num,b[maxn],dc[maxn],root[maxn];
3   int rdc(int x){return lower_bound(dc+1,dc+num+1,x)-dc;}
4   void init()
5   {
6       int i;
7       n=read(); q=read();
8       tr(i,1,n) b[i]=read();
9       memcpy(dc,b,(n+1)*sizeof(int));
10      sort(dc+1,dc+n+1);
11      num=unique(dc+1,dc+n+1)-(dc+1);
12  }
13  int insert(int tx,int l,int r,int x)
14  {
15      int t,mid=(l+r)>>1;
16      a[t=++m]=a[tx]; a[t].size++;
17      if (l==r) return t;
18      if (x<=mid) a[t].l=insert(a[tx].l,l,mid,x);
19      else a[t].r=insert(a[tx].r,mid+1,r,x);
20      return t;
21  }
22  int kth(int tx,int ty,int l,int r,int k)
23  {
24      int ds,mid=(l+r)>>1;
```

```
25      if (l==r) return l;
26      if (k<=(ds=a[a[ty].l].size-a[a[tx].l].size))
27          return kth(a[tx].l,a[ty].l,l,mid,k);
28      else return kth(a[tx].r,a[ty].r,mid+1,r,k-ds);
29  }
30  void work()
31  {
32      int i,x,y,z;
33      tr(i,1,n) root[i]=insert(root[i-1],1,num,rdc(b[i]));
34      tr(i,1,q)
35      {
36          x=read(); y=read(); z=read();
37          writeln(dc[kth(root[x-1],root[y],1,num,z)]);
38      }
39  }
```

7.7 RMQ-ST

$O(NlogN)$ $O(1)$

```
1   //！！注意！！__builtin_clz只有g++能用
2   //x为int时，31-__builtin_clz(x) 等价于 int(log(x)/log(2))
3   //x为ll时，63-__builtin_clzll(x) 等价于 (ll)(log(x)/log(2))
4   int n,q,mn[maxn][maxln];
5   void init()
6   {
7       int i;
8       n=read(); q=read();
9       tr(i,1,n) mn[i][0]=read();
10  }
11  void st()
12  {
13      int i,j,ln;
14      ln=31-__builtin_clz(n);
15      tr(i,1,ln) tr(j,1,n-(1<<i)+1)
16          mn[j][i]=min(mn[j][i-1],mn[j+(1<<(i-1))][i-1]);
17  }
18  void work()
19  {
20      int i,x,y,t;
21      st();
22      tr(i,1,q)
23      {
24          x=read(); y=read();
25          t=31-__builtin_clz(y-x+1);
26          writeln(min(mn[x][t],mn[y-(1<<t)+1][t]));
27      }
28  }
```

8 其它

8.1 n 皇后问题-构造

（输出任一方案），$O(n)$

```
int n,m;
void print(int x){writeb(++m); writeln(x);}
void solve()
{
    int i;
    if (n%6!=2&&n%6!=3)
    {
        for(i=2;i<=n;i+=2) print(i);
        for(i=1;i<=n;i+=2) print(i);
    } else
    {
        int k=n/2;
        if(k%2==0)
        {
            for(i=k;i<=n;i+=2) print(i);
            for(i=2;i<=k-2;i+=2) print(i);
            for(i=k+3;i<=n-1;i+=2) print(i);
            for(i=1;i<=k+1;i+=2) print(i);
            if (n%2==1) print(n);
        } else
        {
            for(i=k;i<=n-1;i+=2) print(i);
            for(i=1;i<=k-2;i+=2) print(i);
            for(i=k+3;i<=n;i+=2) print(i);
            for(i=2;i<=k+1;i+=2) print(i);
            if (n%2==1) print(n);
        }
    }
}
```

9 纯公式/定理

9.1 数学公式

9.1.1 三角

■ 复分析欧拉公式

$$e^{ix} = \cos x + i \sin x$$

（可简单导出棣莫弗定理）

■ 和差公式

$\sin(\alpha \pm \beta) = \sin \alpha \cos \beta \pm \cos \alpha \sin \beta$ $\qquad$ $\cos(\alpha \pm \beta) = \cos \alpha \cos \beta \mp \sin \alpha \sin \beta$

$\tan(\alpha \pm \beta) = \frac{\tan \alpha \pm \tan \beta}{1 \mp \tan \alpha \tan \beta}$

■ 和差化积

$\sin \alpha + \sin \beta = 2 \sin \frac{\alpha+\beta}{2} \cos \frac{\alpha-\beta}{2}$ $\qquad$ $\cos \alpha + \cos \beta = 2 \cos \frac{\alpha+\beta}{2} \cos \frac{\alpha-\beta}{2}$

$\cos \alpha - \cos \beta = -2 \sin \frac{\alpha+\beta}{2} \sin \frac{\alpha-\beta}{2}$ $\qquad$ $\sin \alpha - \sin \beta = 2 \cos \frac{\alpha+\beta}{2} \sin \frac{\alpha-\beta}{2}$

■ 积化和差

$\sin \alpha \sin \beta = \frac{\cos(\alpha-\beta)-\cos(\alpha+\beta)}{2}$ $\qquad$ $\cos \alpha \cos \beta = \frac{\cos(\alpha-\beta)+\cos(\alpha+\beta)}{2}$

$\sin \alpha \cos \beta = \frac{\cos(\alpha+\beta)+\cos(\alpha-\beta)}{2}$ $\qquad$ $\cos \alpha \sin \beta = \frac{\cos(\alpha+\beta)-\cos(\alpha-\beta)}{2}$

■ 二、三、n 倍角（切比雪夫）

$\sin 2\theta = 2 \sin \theta \cos \theta = \frac{2 \tan \theta}{1+\tan^2 \theta}$

$\cos 2\theta = \cos^2 \theta - \sin^2 \theta = 2 \cos^2 \theta - 1 = 1 - 2 \sin^2 \theta = \frac{1-\tan^2 \theta}{1+\tan^2 \theta}$

$\tan 2\theta = \frac{2 \tan \theta}{1-\tan^2 \theta} = \frac{1}{1-\tan \theta} - \frac{1}{1+\tan \theta}$

$\sin 3\theta = 3 \sin \theta - 4 \sin^3 \theta$ $\qquad$ $\cos 3\theta = 4 \cos^3 \theta - 3 \cos \theta$ $\qquad$ $\tan 3\theta = \frac{3 \tan \theta - \tan^3 \theta}{1-3 \tan^2 \theta}$

$$\sin n\theta = \sum_{k=0}^{n} \binom{n}{k} \cos^k \theta \sin^{n-k} \theta \sin \left[\frac{1}{2}(n-k)\pi \right]$$

$$= \sin \theta \sum_{k=0}^{\lfloor \frac{n-1}{2} \rfloor} (-1)^k \binom{n-1-k}{k} (2 \cos \theta)^{n-1-2k}$$

$$\cos n\theta = \sum_{k=0}^{n} \binom{n}{k} \cos^k \theta \sin^{n-k} \theta \cos \left[\frac{1}{2}(n-k)\pi \right]$$

$$= \frac{1}{2} \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} (-1)^k \frac{n}{n-k} \binom{n-k}{k} (2 \cos \theta)^{n-2k}$$

■ 二、三次降幂

$\sin^2 \theta = \frac{1-\cos 2\theta}{2}$ $\qquad$ $\cos^2 \theta = \frac{1+\cos 2\theta}{2}$ $\qquad$ $\sin^3 \theta = \frac{3 \sin \theta - \sin 3\theta}{4}$ $\qquad$ $\cos^3 \theta = \frac{3 \cos \theta + \cos 3\theta}{4}$

■ 万能公式

$t = \tan \frac{\theta}{2}$ $\Rightarrow$ $\sin \theta = \frac{2t}{1+t^2}$ $\qquad$ $\cos \theta = \frac{1-t^2}{1+t^2}$ $\qquad$ $\sin \theta = \frac{2t}{1-t^2}$ $\qquad$ $\mathrm{d}x = \frac{2}{1+t^2} \mathrm{d}t$

■ 连乘

$$\prod_{k=0}^{n-1} \cos 2^k \theta = \frac{\sin 2^n \theta}{2^n \sin \theta} \qquad \prod_{k=0}^{n-1} \sin \left(x + \frac{k\pi}{n} \right) = \frac{\sin nx}{2^{n-1}}$$

$$\prod_{k=1}^{n-1} \sin \left(\frac{k\pi}{n} \right) = \frac{n}{2^{n-1}} \qquad \prod_{k=1}^{n-1} \sin \left(\frac{k\pi}{2n} \right) = \frac{\sqrt{n}}{2^{n-1}} \qquad \prod_{k=1}^{n} \sin \left(\frac{k\pi}{2n+1} \right) = \frac{\sqrt{2n+1}}{2^n}$$

$$\prod_{k=1}^{n-1}\cos\left(\frac{k\pi}{n}\right)=\frac{\sin\frac{n\pi}{2}}{2^{n-1}}\quad\prod_{k=1}^{n-1}\cos\left(\frac{k\pi}{2n}\right)=\frac{\sqrt{n}}{2^{n-1}}\quad\prod_{k=1}^{n}\cos\left(\frac{k\pi}{2n+1}\right)=\frac{1}{2^n}$$

$$\prod_{k=1}^{n-1}\tan\left(\frac{k\pi}{n}\right)=\frac{n}{\sin\frac{n\pi}{2}}\quad\prod_{k=1}^{n-1}\tan\left(\frac{k\pi}{2n}\right)=1\quad\prod_{k=1}^{n}\tan\frac{k\pi}{2n+1}=\sqrt{2n+1}$$

■ 其它

$$x+y+z=n\pi\Rightarrow\tan x+\tan y+\tan z=\tan x\tan y\tan z$$
$$x+y+z=n\pi+\frac{\pi}{2}\Rightarrow\cot x+\cot y+\cot z=\cot x\cot y\cot z$$
$$x+y+z=\pi\Rightarrow\sin 2x+\sin 2y+\sin 2z=4\sin x\sin y\sin z$$
$$\sin(x+y)\sin(x-y)=\sin^2 x-\sin^2 y=\cos^2 y-\cos^2 x$$
$$\cos(x+y)\cos(x-y)=\cos^2 x-\sin^2 y=\cos^2 y-\sin^2 x$$

9.1.2 重要数与数列

■ 幂级数

$$\sum_{i=1}^{n}i=\frac{1}{2}n(n+1)\quad\sum_{i=1}^{n}i^2=\frac{1}{3}n(n+\frac{1}{2})(n+1)\quad\sum_{i=1}^{n}i^3=(\sum_{i=1}^{n}i)^2=\frac{1}{4}n^2(n+1)^2$$

$$\sum_{i=1}^{n}i^m=\frac{1}{m+1}\sum_{k=0}^{m}\binom{m+1}{k}b_k(n+1)^{m+1-k}$$

$$=\frac{1}{m+1}\left[(n+1)^{m+1}-1-\sum_{i=1}^{n}\left((i+1)^{m+1}-i^{m+1}-(m+1)i^m\right)\right]$$

■ 几何级数

$$\sum_{i=0}^{n}ic^i=\frac{nc^{n+2}-(n+1)c^{n+1}+c}{(c-1)^2},\quad c\neq 1\qquad\sum_{i=0}^{\infty}ic^i=\frac{c}{(1-c)^2},\quad|c|<1$$

■ 调和级数

H_n 表调和级数，

$$H_n=\sum_{k=1}^{n}\frac{1}{k}$$

$$\sum_{i=1}^{n}iH_i=\frac{n(n+1)}{2}H_n-\frac{n(n-1)}{4}\qquad\sum_{i=1}^{n}H_i=(n+1)H_n-n,$$

$$\sum_{i=1}^{n}\binom{i}{m}H_i=\binom{n+1}{m+1}\left(H_{n+1}-\frac{1}{m+1}\right)$$

■ 组合数

C(i,j)	0	1	2	3	4	5	6	7	8	9	10	11
0	1											
1	1	1										
2	1	2	1									
3	1	3	3	1								
4	1	4	6	4	1							
5	1	5	10	10	5	1						
6	1	6	15	20	15	6	1					
7	1	7	21	35	35	21	7	1				
8	1	8	28	56	70	56	28	8	1			
9	1	9	36	84	126	126	84	36	9	1		
10	1	10	45	120	210	252	210	120	45	10	1	
11	1	11	55	165	330	462	462	330	165	55	11	1

$$\binom{n}{k}=\binom{n}{n-k}=\frac{n}{k}\binom{n-1}{k-1}=\binom{n-1}{k}+\binom{n-1}{k-1}\qquad\binom{n}{m}\binom{m}{k}=\binom{n}{k}\binom{n-k}{m-k}$$

$$\sum_{k=0}^{n}\binom{r+k}{k}=\binom{r+n+1}{n}\qquad\sum_{k=0}^{n}\binom{k}{m}=\binom{n+1}{m+1}$$

$$\sum_{k=0}^{n}\binom{r}{k}\binom{s}{n-k}=\binom{r+s}{n}$$

■ 第一类斯特林数

$\begin{bmatrix}n\\k\end{bmatrix}$ 表第一类斯特林数，表 n 元素分作 k 个环排列的方法数，

$$\begin{bmatrix}n\\0\end{bmatrix}=0,\begin{bmatrix}1\\1\end{bmatrix}=1,\begin{bmatrix}n\\k\end{bmatrix}=\begin{bmatrix}n-1\\k-1\end{bmatrix}+(n-1)\begin{bmatrix}n-1\\k\end{bmatrix}$$

s(i,j)	1	2	3	4	5	6	7	8
1	1							
2	1	1						
3	2	3	1					
4	6	11	6	1				
5	24	50	35	10	1			
6	120	274	225	85	15	1		
7	720	1764	1624	735	175	21	1	
8	5040	13068	13132	6769	1960	322	28	1

$$\begin{bmatrix}n\\1\end{bmatrix}=(n-1)!\qquad\begin{bmatrix}n\\2\end{bmatrix}=(n-1)!H_{n-1}\qquad\begin{bmatrix}n\\n-1\end{bmatrix}=\binom{n}{2}\qquad\sum_{k=0}^{n}\begin{bmatrix}n\\k\end{bmatrix}=n!$$

$$\begin{bmatrix}n+1\\m+1\end{bmatrix}=\sum_{k}\begin{bmatrix}n\\k\end{bmatrix}\binom{k}{m}=n!\sum_{k=0}^{n}\frac{1}{k!}\begin{bmatrix}k\\m\end{bmatrix}$$

$$\begin{bmatrix} n \\ m \end{bmatrix} = \sum_k \begin{bmatrix} n+1 \\ k+1 \end{bmatrix} \binom{k}{m} (-1)^{m-k} \qquad \begin{bmatrix} m+n+1 \\ m \end{bmatrix} = \sum_{k=0}^{m} k(n+k) \begin{bmatrix} n+k \\ k \end{bmatrix}$$

$$\begin{bmatrix} n \\ \ell+m \end{bmatrix} \binom{\ell+m}{\ell} = \sum_k \begin{bmatrix} k \\ \ell \end{bmatrix} \begin{bmatrix} n-k \\ m \end{bmatrix} \binom{n}{k}$$

■ 第二类斯特林数

$\left\{ \begin{matrix} n \\ k \end{matrix} \right\}$ 表第二类斯特林数，表基数为 n 的集合的 k 份划分方法数，

$$\left\{ \begin{matrix} n \\ 1 \end{matrix} \right\} = \left\{ \begin{matrix} n \\ n \end{matrix} \right\} = 1, \left\{ \begin{matrix} n \\ k \end{matrix} \right\} = \left\{ \begin{matrix} n-1 \\ k-1 \end{matrix} \right\} + k \left\{ \begin{matrix} n-1 \\ k \end{matrix} \right\}$$

S(i,j)	1	2	3	4	5	6	7	8
1	1							
2	1	1						
3	1	3	1					
4	1	7	6	1				
5	1	15	25	10	1			
6	1	31	90	65	15	1		
7	1	63	301	350	140	21	1	
8	1	127	966	1701	1050	266	28	1

$$\left\{ \begin{matrix} n \\ 2 \end{matrix} \right\} = 2^{n-1} - 1 \qquad \left\{ \begin{matrix} n \\ n-1 \end{matrix} \right\} = \binom{n}{2} \qquad \begin{bmatrix} n \\ k \end{bmatrix} \geq \left\{ \begin{matrix} n \\ k \end{matrix} \right\}$$

$$\left\{ \begin{matrix} n+1 \\ m+1 \end{matrix} \right\} = \sum_k \binom{n}{k} \left\{ \begin{matrix} k \\ m \end{matrix} \right\} = \sum_{k=0}^{n} \left\{ \begin{matrix} k \\ m \end{matrix} \right\} (m+1)^{n-k}$$

$$\left\{ \begin{matrix} n \\ m \end{matrix} \right\} = \sum_k \binom{n}{k} \left\{ \begin{matrix} k+1 \\ m+1 \end{matrix} \right\} (-1)^{n-k} \qquad \left\{ \begin{matrix} m+n+1 \\ m \end{matrix} \right\} = \sum_{k=0}^{m} k \left\{ \begin{matrix} n+k \\ k \end{matrix} \right\}$$

$$\binom{n}{m} = \sum_k \left\{ \begin{matrix} n+1 \\ k+1 \end{matrix} \right\} \begin{bmatrix} k \\ m \end{bmatrix} (-1)^{m-k}$$

$$(n-m)! \binom{n}{m} = \sum_k \begin{bmatrix} n+1 \\ k+1 \end{bmatrix} \left\{ \begin{matrix} k \\ m \end{matrix} \right\} (-1)^{m-k}, \quad \forall n \geq m$$

$$\left\{ \begin{matrix} n \\ n-m \end{matrix} \right\} = \sum_k \binom{m-n}{m+k} \binom{m+n}{n+k} \begin{bmatrix} m+k \\ k \end{bmatrix}$$

$$\begin{bmatrix} n \\ n-m \end{bmatrix} = \sum_k \binom{m-n}{m+k} \binom{m+n}{n+k} \left\{ \begin{matrix} m+k \\ k \end{matrix} \right\}$$

$$\left\{ \begin{matrix} n \\ \ell+m \end{matrix} \right\} \binom{\ell+m}{\ell} = \sum_k \left\{ \begin{matrix} k \\ \ell \end{matrix} \right\} \left\{ \begin{matrix} n-k \\ m \end{matrix} \right\} \binom{n}{k}$$

■ 贝尔数

B_n 表贝尔数，表基数为 n 的集合的划分方法数，

$$B_0 = 1, B_{n+1} = \sum_{k=0}^{n} \binom{n}{k} B_k$$

n	0	1	2	3	4	5	6	7	8	9	10	11
B_n	1	1	2	5	15	52	203	877	4140	21147	115975	678570

$$B_n = \sum_{k=1}^{n} \left\{ \begin{matrix} n \\ k \end{matrix} \right\} \qquad B_n = \frac{1}{e} \sum_{k=0}^{\infty} \frac{k^n}{k!} \qquad \sum_{n=0}^{\infty} \frac{B_n}{n!} x^n = e^{e^x - 1}$$

p 是质数 $\Rightarrow B_{n+p} \equiv B_n + B_{n+1} \pmod{p}$

■ 卡特兰数

C_n 表卡特兰数，

$$C_n = \frac{1}{n+1} \binom{2n}{n} \quad n \geq 0$$

n	0	1	2	3	4	5	6	7	8	9	10	11
C_n	1	1	2	5	14	42	132	429	1430	4862	16796	58786

$$C_n = \binom{2n}{n} - \binom{2n}{n+1} \quad \forall n \geq 1 \qquad C_{n+1} = \sum_{k=0}^{n} C_k C_{n-k} \quad \forall n \geq 0$$

$C_{n+1} = \frac{4n+2}{n+2} C_n \qquad C_n$ 为奇数 $\Leftrightarrow n = 2^k - 1, k \in \mathbb{Q}$

大小为 n 的不同构二叉树数目为 C_n；$n \times n$ 格点不越过对角线的单调路径（比如仅向右或上）数目为 C_n；$n+2$ 边凸多边形分成三角形的方法数为 C_n；高度为 n 的阶梯形分成 n 个长方形的方法数为 C_n；待进栈的 n 个元素的出栈序列种数为 C_n

■ 伯努利数

b_n 表 n 次伯努利数，

$$b_0 = 1, \sum_{k=0}^{m} \binom{m+1}{k} b_k = 0$$

n	0	1	2	3	4	5	6	7	8	9	10	11	12
b_n	1	$-\frac{1}{2}$	$\frac{1}{6}$	0	$-\frac{1}{30}$	0	$\frac{1}{42}$	0	$-\frac{1}{30}$	0	$\frac{5}{66}$	0	$-\frac{691}{2730}$

■ 斐波那契数列

F_n 表斐波那契数列，$F_0 = 0, F_1 = F_2 = 1, F_n = F_{n-1} + F_{n-2}$

9.1.3 泰勒级数

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!} (x-a)^n$$

$$\frac{1}{1-x} = \sum_{i=0}^{\infty} x^i \qquad \frac{1}{1-cx} = \sum_{i=0}^{\infty} c^i x^i \qquad \frac{1}{1-x^n} = \sum_{i=0}^{\infty} x^{ni} \qquad \frac{x}{(1-x)^2} = \sum_{i=0}^{\infty} i x^i$$

$$\sum_{k=0}^{n} \left\{ {n \atop k} \right\} \frac{k! z^k}{(1-z)^{k+1}} = \sum_{i=0}^{\infty} i^n x^i \qquad\qquad e^x = \sum_{i=0}^{\infty} \frac{x^i}{i!}$$

$$\ln(1+x) = \sum_{i=1}^{\infty} (-1)^{i+1} \frac{x^i}{i} \qquad\qquad \ln\frac{1}{1-x} = \sum_{i=1}^{\infty} \frac{x^i}{i}$$

$$\sin x = \sum_{i=0}^{\infty} (-1)^i \frac{x^{2i+1}}{(2i+1)!} \qquad\qquad \cos x = \sum_{i=0}^{\infty} (-1)^i \frac{x^{2i}}{(2i)!}$$

$$\tan^{-1} x = \sum_{i=0}^{\infty} (-1)^i \frac{x^{2i+1}}{(2i+1)} \qquad\qquad (1+x)^n = \sum_{i=0}^{\infty} \binom{n}{i} x^i$$

$$\frac{1}{(1-x)^{n+1}} = \sum_{i=0}^{\infty} \binom{i+n}{i} x^i \qquad\qquad \frac{x}{e^x - 1} = \sum_{i=0}^{\infty} \frac{b_i x^i}{i!}$$

$$\frac{1}{2x}(1 - \sqrt{1-4x}) = \sum_{i=0}^{\infty} \frac{1}{i+1} \binom{2i}{i} x^i \qquad \frac{1}{\sqrt{1-4x}} = \sum_{i=0}^{\infty} \binom{2i}{i} x^i$$

$$\frac{1}{\sqrt{1-4x}} \left(\frac{1 - \sqrt{1-4x}}{2x} \right)^n = \sum_{i=0}^{\infty} \binom{2i+n}{i} x^i \qquad \frac{1}{1-x} \ln\frac{1}{1-x} = \sum_{i=1}^{\infty} H_i x^i$$

$$\frac{1}{2} \left(\ln\frac{1}{1-x} \right)^2 = \sum_{i=2}^{\infty} \frac{H_{i-1} x^i}{i} \qquad\qquad \frac{x}{1-x-x^2} = \sum_{i=0}^{\infty} F_i x^i$$

$$\frac{F_n x}{1 - (F_{n-1} + F_{n+1})x - (-1)^n x^2} = \sum_{i=0}^{\infty} F_{ni} x^i.$$

$$\frac{1}{(1-x)^{n+1}} \ln\frac{1}{1-x} = \sum_{i=0}^{\infty} (H_{n+i} - H_n) \binom{n+i}{i} x^i, \left(\frac{1}{x} \right)^{\overline{-n}} = \sum_{i=0}^{\infty} \left\{ {i \atop n} \right\} x^i$$

$$x^{\overline{n}} = \sum_{i=0}^{\infty} \left[{n \atop i} \right] x^i, (e^x - 1)^n = \sum_{i=0}^{\infty} \left\{ {i \atop n} \right\} \frac{n! x^i}{i!}$$

$$\left(\ln\frac{1}{1-x} \right)^n = \sum_{i=0}^{\infty} \left[{i \atop n} \right] \frac{n! x^i}{i!}, x \cot x = \sum_{i=0}^{\infty} \frac{(-4)^i b_{2i} x^{2i}}{(2i)!}$$

$$\tan x = \sum_{i=1}^{\infty} (-1)^{i-1} \frac{2^{2i}(2^{2i}-1) b_{2i} x^{2i-1}}{(2i)!}, \zeta(x) = \sum_{i=1}^{\infty} \frac{1}{i^x}$$

$$\frac{1}{\zeta(x)} = \sum_{i=1}^{\infty} \frac{\mu(i)}{i^x}, \frac{\zeta(x-1)}{\zeta(x)} = \sum_{i=1}^{\infty} \frac{\phi(i)}{i^x}$$

9.1.4 导数

■ 几个导数

$(\tan x)' = \sec^2 x \quad (\arctan x)' = \frac{1}{1+x^2} \quad (\arcsin x)' = \frac{1}{\sqrt{1-x^2}} \quad (\arccos x)' = -\frac{1}{\sqrt{1-x^2}}$

$(\sinh x)' = \cosh x = \frac{e^x + e^{-x}}{2} \quad (\cosh x)' = \sinh x = \frac{e^x - e^{-x}}{2}$

■ 高阶导数
(莱布尼茨公式)

$$(uv)^{(n)} = \sum_{k=0}^{n} \binom{n}{k} u^{(n-k)} v^{(k)}$$

$$(x^a)^{(n)} = x^{a-n} \prod_{k=0}^{n-1}(a-k) \qquad (\tfrac{1}{x})^{(n)} = (-1)^n \frac{n!}{x^{n+1}}$$

$(a^x)^{(n)} = a^x \ln^n a \ (a>0) \qquad (\ln x)^{(n)} = (-1)^{n-1} \frac{(n-1)!}{x^n}$

$(\sin(kx+b))^{(n)} = k^n \sin(kx + b + \frac{n\pi}{2}) \qquad (\cos(kx+b))^{(n)} = k^n \cos(kx + b + \frac{n\pi}{2})$

9.1.5 积分表

■ $ax + b(a \neq 0)$

1. $\int \frac{dx}{ax+b} = \frac{1}{a} \ln|ax+b| + C$

2. $\int (ax+b)^\mu dx = \frac{1}{a(\mu+1)} (ax+b)^{\mu+1} + C (\mu \neq 1)$

3. $\int \frac{x}{ax+b} dx = \frac{1}{a^2}(ax+b - b\ln|ax+b|) + C$

4. $\int \frac{x^2}{ax+b} dx = \frac{1}{a^3} \left(\frac{1}{2}(ax+b)^2 - 2b(ax+b) + b^2 \ln|ax+b| \right) + C$

5. $\int \frac{dx}{x(ax+b)} = -\frac{1}{b} \ln \left| \frac{ax+b}{x} \right| + C$

6. $\int \frac{dx}{x^2(ax+b)} = -\frac{1}{bx} + \frac{a}{b^2} \ln \left| \frac{ax+b}{x} \right| + C$

7. $\int \frac{x}{(ax+b)^2} dx = \frac{1}{a^2} \left(\ln|ax+b| + \frac{b}{ax+b} \right) + C$

8. $\int \frac{x^2}{(ax+b)^2} dx = \frac{1}{a^3} \left(ax+b - 2b\ln|ax+b| - \frac{b^2}{ax+b} \right) + C$

9. $\int \frac{dx}{x(ax+b)^2} = \frac{1}{b(ax+b)} - \frac{1}{b^2} \ln \left| \frac{ax+b}{x} \right| + C$

■ $\sqrt{ax+b}$

1. $\int \sqrt{ax+b} dx = \frac{2}{3a} \sqrt{(ax+b)^3} + C$

2. $\int x\sqrt{ax+b} dx = \frac{2}{15a^2}(3ax - 2b)\sqrt{(ax+b)^3} + C$

3. $\int x^2 \sqrt{ax+b} dx = \frac{2}{105a^3}(15a^2 x^2 - 12abx + 8b^2)\sqrt{(ax+b)^3} + C$

4. $\int \frac{x}{\sqrt{ax+b}} dx = \frac{2}{3a^2}(ax - 2b)\sqrt{ax+b} + C$

5. $\int \frac{x^2}{\sqrt{ax+b}} dx = \frac{2}{15a^3}(3a^2 x^2 - 4abx + 8b^2)\sqrt{ax+b} + C$

6. $\int \frac{\mathrm{d}x}{x\sqrt{ax+b}} = \begin{cases} \frac{1}{\sqrt{b}} \ln \left| \frac{\sqrt{ax+b}-\sqrt{b}}{\sqrt{ax+b}+\sqrt{b}} \right| + C & (b > 0) \\ \frac{2}{\sqrt{-b}} \arctan \sqrt{\frac{ax+b}{-b}} + C & (b < 0) \end{cases}$

7. $\int \frac{\mathrm{d}x}{x^2\sqrt{ax+b}} = -\frac{\sqrt{ax+b}}{bx} - \frac{a}{2b} \int \frac{\mathrm{d}x}{x\sqrt{ax+b}}$

8. $\int \frac{\sqrt{ax+b}}{x}\mathrm{d}x = 2\sqrt{ax+b} + b \int \frac{\mathrm{d}x}{x\sqrt{ax+b}}$

9. $\int \frac{\sqrt{ax+b}}{x^2}\mathrm{d}x = -\frac{\sqrt{ax+b}}{x} + \frac{a}{2} \int \frac{\mathrm{d}x}{x\sqrt{ax+b}}$

■ $x^2 \pm a^2$

1. $\int \frac{\mathrm{d}x}{x^2+a^2} = \frac{1}{a} \arctan \frac{x}{a} + C$

2. $\int \frac{\mathrm{d}x}{(x^2+a^2)^n} = \frac{x}{2(n-1)a^2(x^2+a^2)^{n-1}} + \frac{2n-3}{2(n-1)a^2} \int \frac{\mathrm{d}x}{(x^2+a^2)^{n-1}}$

3. $\int \frac{\mathrm{d}x}{x^2-a^2} = \frac{1}{2a} \ln \left| \frac{x-a}{x+a} \right| + C$

■ $ax^2 + b(a > 0)$

1. $\int \frac{\mathrm{d}x}{ax^2+b} = \begin{cases} \frac{1}{\sqrt{ab}} \arctan \sqrt{\frac{a}{b}}x + C & (b > 0) \\ \frac{1}{2\sqrt{-ab}} \ln \left| \frac{\sqrt{a}x-\sqrt{-b}}{\sqrt{a}x+\sqrt{-b}} \right| + C & (b < 0) \end{cases}$

2. $\int \frac{x}{ax^2+b}\mathrm{d}x = \frac{1}{2a} \ln |ax^2 + b| + C$

3. $\int \frac{x^2}{ax^2+b}\mathrm{d}x = \frac{x}{a} - \frac{b}{a} \int \frac{\mathrm{d}x}{ax^2+b}$

4. $\int \frac{\mathrm{d}x}{x(ax^2+b)} = \frac{1}{2b} \ln \frac{x^2}{|ax^2+b|} + C$

5. $\int \frac{\mathrm{d}x}{x^2(ax^2+b)} = -\frac{1}{bx} - \frac{a}{b} \int \frac{\mathrm{d}x}{ax^2+b}$

6. $\int \frac{\mathrm{d}x}{x^3(ax^2+b)} = \frac{a}{2b^2} \ln \frac{|ax^2+b|}{x^2} - \frac{1}{2bx^2} + C$

7. $\int \frac{\mathrm{d}x}{(ax^2+b)^2} = \frac{x}{2b(ax^2+b)} + \frac{1}{2b} \int \frac{\mathrm{d}x}{ax^2+b}$

■ $ax^2 + bx + c(a > 0)$

1. $\frac{\mathrm{d}x}{ax^2+bx+c} = \begin{cases} \frac{2}{\sqrt{4ac-b^2}} \arctan \frac{2ax+b}{\sqrt{4ac-b^2}} + C & (b^2 < 4ac) \\ \frac{1}{\sqrt{b^2-4ac}} \ln \left| \frac{2ax+b-\sqrt{b^2-4ac}}{2ax+b+\sqrt{b^2-4ac}} \right| + C & (b^2 > 4ac) \end{cases}$

2. $\int \frac{x}{ax^2+bx+c}\mathrm{d}x = \frac{1}{2a} \ln |ax^2 + bx + c| - \frac{b}{2a} \int \frac{\mathrm{d}x}{ax^2+bx+c}$

■ $\sqrt{x^2 + a^2}(a > 0)$

1. $\int \frac{\mathrm{d}x}{\sqrt{x^2+a^2}} = \operatorname{arsh}\frac{x}{a} + C_1 = \ln(x + \sqrt{x^2 + a^2}) + C$

2. $\int \frac{\mathrm{d}x}{\sqrt{(x^2+a^2)^3}} = \frac{x}{a^2\sqrt{x^2+a^2}} + C$

3. $\int \frac{x}{\sqrt{x^2+a^2}}\mathrm{d}x = \sqrt{x^2 + a^2} + C$

4. $\int \frac{x}{\sqrt{(x^2+a^2)^3}}\mathrm{d}x = -\frac{1}{\sqrt{x^2+a^2}} + C$

5. $\int \frac{x^2}{\sqrt{x^2+a^2}}\mathrm{d}x = \frac{x}{2}\sqrt{x^2 + a^2} - \frac{a^2}{2} \ln(x + \sqrt{x^2 + a^2}) + C$

6. $\int \frac{x^2}{\sqrt{(x^2+a^2)^3}}\mathrm{d}x = -\frac{x}{\sqrt{x^2+a^2}} + \ln(x + \sqrt{x^2 + a^2}) + C$

7. $\int \frac{\mathrm{d}x}{x\sqrt{x^2+a^2}} = \frac{1}{a} \ln \frac{\sqrt{x^2+a^2}-a}{|x|} + C$

8. $\int \frac{\mathrm{d}x}{x^2\sqrt{x^2+a^2}} = -\frac{\sqrt{x^2+a^2}}{a^2x} + C$

9. $\int \sqrt{x^2 + a^2}\mathrm{d}x = \frac{x}{2}\sqrt{x^2 + a^2} + \frac{a^2}{2} \ln(x + \sqrt{x^2 + a^2}) + C$

10. $\int \sqrt{(x^2 + a^2)^3}\mathrm{d}x = \frac{x}{8}(2x^2 + 5a^2)\sqrt{x^2 + a^2} + \frac{3}{8}a^4 \ln(x + \sqrt{x^2 + a^2}) + C$

11. $\int x\sqrt{x^2 + a^2}\mathrm{d}x = \frac{1}{3}\sqrt{(x^2 + a^2)^3} + C$

12. $\int x^2\sqrt{x^2 + a^2}\mathrm{d}x = \frac{x}{8}(2x^2 + a^2)\sqrt{x^2 + a^2} - \frac{a^4}{8} \ln(x + \sqrt{x^2 + a^2}) + C$

13. $\int \frac{\sqrt{x^2+a^2}}{x}\mathrm{d}x = \sqrt{x^2 + a^2} + a \ln \frac{\sqrt{x^2+a^2}-a}{|x|} + C$

14. $\int \frac{\sqrt{x^2+a^2}}{x^2}\mathrm{d}x = -\frac{\sqrt{x^2+a^2}}{x} + \ln(x + \sqrt{x^2 + a^2}) + C$

■ $\sqrt{x^2 - a^2}(a > 0)$

1. $\int \frac{\mathrm{d}x}{\sqrt{x^2-a^2}} = \frac{x}{|x|}\operatorname{arch}\frac{|x|}{a} + C_1 = \ln \left| x + \sqrt{x^2 - a^2} \right| + C$

2. $\int \frac{\mathrm{d}x}{\sqrt{(x^2-a^2)^3}} = -\frac{x}{a^2\sqrt{x^2-a^2}} + C$

3. $\int \frac{x}{\sqrt{x^2-a^2}}\mathrm{d}x = \sqrt{x^2 - a^2} + C$

4. $\int \frac{x}{\sqrt{(x^2-a^2)^3}}\mathrm{d}x = -\frac{1}{\sqrt{x^2-a^2}} + C$

5. $\int \frac{x^2}{\sqrt{x^2-a^2}}\mathrm{d}x = \frac{x}{2}\sqrt{x^2 - a^2} + \frac{a^2}{2} \ln |x + \sqrt{x^2 - a^2}| + C$

6. $\int \frac{x^2}{\sqrt{(x^2-a^2)^3}}\mathrm{d}x = -\frac{x}{\sqrt{x^2-a^2}} + \ln |x + \sqrt{x^2 - a^2}| + C$

7. $\int \frac{\mathrm{d}x}{x\sqrt{x^2-a^2}} = \frac{1}{a}\arccos\frac{a}{|x|} + C$

8. $\int \frac{\mathrm{d}x}{x^2\sqrt{x^2-a^2}} = \frac{\sqrt{x^2-a^2}}{a^2 x} + C$

9. $\int \sqrt{x^2-a^2}\mathrm{d}x = \frac{x}{2}\sqrt{x^2-a^2} - \frac{a^2}{2}\ln|x+\sqrt{x^2-a^2}| + C$

10. $\int \sqrt{(x^2-a^2)^3}\mathrm{d}x = \frac{x}{8}(2x^2-5a^2)\sqrt{x^2-a^2} + \frac{3}{8}a^4\ln|x+\sqrt{x^2-a^2}| + C$

11. $\int x\sqrt{x^2-a^2}\mathrm{d}x = \frac{1}{3}\sqrt{(x^2-a^2)^3} + C$

12. $\int x^2\sqrt{x^2-a^2}\mathrm{d}x = \frac{x}{8}(2x^2-a^2)\sqrt{x^2-a^2} - \frac{a^4}{8}\ln|x+\sqrt{x^2-a^2}| + C$

13. $\int \frac{\sqrt{x^2-a^2}}{x}\mathrm{d}x = \sqrt{x^2-a^2} - a\arccos\frac{a}{|x|} + C$

14. $\int \frac{\sqrt{x^2-a^2}}{x^2}\mathrm{d}x = -\frac{\sqrt{x^2-a^2}}{x} + \ln|x+\sqrt{x^2-a^2}| + C$

■ $\sqrt{a^2-x^2}(a>0)$

1. $\int \frac{\mathrm{d}x}{\sqrt{a^2-x^2}} = \arcsin\frac{x}{a} + C$

2. $\frac{\mathrm{d}x}{\sqrt{(a^2-x^2)^3}} = \frac{x}{a^2\sqrt{a^2-x^2}} + C$

3. $\int \frac{x}{\sqrt{a^2-x^2}}\mathrm{d}x = -\sqrt{a^2-x^2} + C$

4. $\int \frac{x}{\sqrt{(a^2-x^2)^3}}\mathrm{d}x = \frac{1}{\sqrt{a^2-x^2}} + C$

5. $\int \frac{x^2}{\sqrt{a^2-x^2}}\mathrm{d}x = -\frac{x}{2}\sqrt{a^2-x^2} + \frac{a^2}{2}\arcsin\frac{x}{a} + C$

6. $\int \frac{x^2}{\sqrt{(a^2-x^2)^3}}\mathrm{d}x = \frac{x}{\sqrt{a^2-x^2}} - \arcsin\frac{x}{a} + C$

7. $\int \frac{\mathrm{d}x}{x\sqrt{a^2-x^2}} = \frac{1}{a}\ln\frac{a-\sqrt{a^2-x^2}}{|x|} + C$

8. $\int \frac{\mathrm{d}x}{x^2\sqrt{a^2-x^2}} = -\frac{\sqrt{a^2-x^2}}{a^2 x} + C$

9. $\int \sqrt{a^2-x^2}\mathrm{d}x = \frac{x}{2}\sqrt{a^2-x^2} + \frac{a^2}{2}\arcsin\frac{x}{a} + C$

10. $\int \sqrt{(a^2-x^2)^3}\mathrm{d}x = \frac{x}{8}(5a^2-2x^2)\sqrt{a^2-x^2} + \frac{3}{8}a^4\arcsin\frac{x}{a} + C$

11. $\int x\sqrt{a^2-x^2}\mathrm{d}x = -\frac{1}{3}\sqrt{(a^2-x^2)^3} + C$

12. $\int x^2\sqrt{a^2-x^2}\mathrm{d}x = \frac{x}{8}(2x^2-a^2)\sqrt{a^2-x^2} + \frac{a^4}{8}\arcsin\frac{x}{a} + C$

13. $\int \frac{\sqrt{a^2-x^2}}{x}\mathrm{d}x = \sqrt{a^2-x^2} + a\ln\frac{a-\sqrt{a^2-x^2}}{|x|} + C$

14. $\int \frac{\sqrt{a^2-x^2}}{x^2}\mathrm{d}x = -\frac{\sqrt{a^2-x^2}}{x} - \arcsin\frac{x}{a} + C$

■ $\sqrt{\pm ax^2+bx+c}(a>0)$

1. $\int \frac{\mathrm{d}x}{\sqrt{ax^2+bx+c}} = \frac{1}{\sqrt{a}}\ln|2ax+b+2\sqrt{a}\sqrt{ax^2+bx+c}| + C$

2. $\int \sqrt{ax^2+bx+c}\mathrm{d}x = \frac{2ax+b}{4a}\sqrt{ax^2+bx+c} + \frac{4ac-b^2}{8\sqrt{a^3}}\ln|2ax+b+2\sqrt{a}\sqrt{ax^2+bx+c}| + C$

3. $\int \frac{x}{\sqrt{ax^2+bx+c}}\mathrm{d}x = \frac{1}{a}\sqrt{ax^2+bx+c} - \frac{b}{2\sqrt{a^3}}\ln|2ax+b+2\sqrt{a}\sqrt{ax^2+bx+c}| + C$

4. $\int \frac{\mathrm{d}x}{\sqrt{c+bx-ax^2}} = -\frac{1}{\sqrt{a}}\arcsin\frac{2ax-b}{\sqrt{b^2+4ac}} + C$

5. $\int \sqrt{c+bx-ax^2}\mathrm{d}x = \frac{2ax-b}{4a}\sqrt{c+bx-ax^2} + \frac{b^2+4ac}{8\sqrt{a^3}}\arcsin\frac{2ax-b}{\sqrt{b^2+4ac}} + C$

6. $\int \frac{x}{\sqrt{c+bx-ax^2}}\mathrm{d}x = -\frac{1}{a}\sqrt{c+bx-ax^2} + \frac{b}{2\sqrt{a^3}}\arcsin\frac{2ax-b}{\sqrt{b^2+4ac}} + C$

■ $\sqrt{\pm\frac{x-a}{x-b}}$ **或** $\sqrt{(x-a)(x-b)}$

1. $\int \sqrt{\frac{x-a}{x-b}}\mathrm{d}x = (x-b)\sqrt{\frac{x-a}{x-b}} + (b-a)\ln(\sqrt{|x-a|}+\sqrt{|x-b|}) + C$

2. $\int \sqrt{\frac{x-a}{b-x}}\mathrm{d}x = (x-b)\sqrt{\frac{x-a}{b-x}} + (b-a)\arcsin\sqrt{\frac{x-a}{b-x}} + C$

3. $\int \frac{\mathrm{d}x}{\sqrt{(x-a)(b-x)}} = 2\arcsin\sqrt{\frac{x-a}{b-x}} + C \ (a<b)$

4. $\int \sqrt{(x-a)(b-x)}\mathrm{d}x = \frac{2x-a-b}{4}\sqrt{(x-a)(b-x)} + \frac{(b-a)^2}{4}\arcsin\sqrt{\frac{x-a}{b-x}} + C, (a<b)$

■ **指数**

1. $\int a^x\mathrm{d}x = \frac{1}{\ln a}a^x + C$

2. $\int \mathrm{e}^{ax}\mathrm{d}x = \frac{1}{a}a^{ax} + C$

3. $\int x\mathrm{e}^{ax}\mathrm{d}x = \frac{1}{a^2}(ax-1)a^{ax} + C$

4. $\int x^n\mathrm{e}^{ax}\mathrm{d}x = \frac{1}{a}x^n\mathrm{e}^{ax} - \frac{n}{a}\int x^{n-1}\mathrm{e}^{ax}\mathrm{d}x$

5. $\int xa^x\mathrm{d}x = \frac{x}{\ln a}a^x - \frac{1}{(\ln a)^2}a^x + C$

6. $\int x^n a^x\mathrm{d}x = \frac{1}{\ln a}x^n a^x - \frac{n}{\ln a}\int x^{n-1}a^x\mathrm{d}x$

7. $\int \mathrm{e}^{ax}\sin bx\mathrm{d}x = \frac{1}{a^2+b^2}\mathrm{e}^{ax}(a\sin bx - b\cos bx) + C$

8. $\int \mathrm{e}^{ax}\cos bx\mathrm{d}x = \frac{1}{a^2+b^2}\mathrm{e}^{ax}(b\sin bx + a\cos bx) + C$

9. $\int \mathrm{e}^{ax}\sin^n bx\mathrm{d}x = \frac{1}{a^2+b^2n^2}\mathrm{e}^{ax}\sin^{n-1}bx(a\sin bx - nb\cos bx) + \frac{n(n-1)b^2}{a^2+b^2n^2}\int \mathrm{e}^{ax}\sin^{n-2}bx\mathrm{d}x$

10. $\int \mathrm{e}^{ax}\cos^n bx\mathrm{d}x = \frac{1}{a^2+b^2n^2}\mathrm{e}^{ax}\cos^{n-1}bx(a\cos bx + nb\sin bx) + \frac{n(n-1)b^2}{a^2+b^2n^2}\int \mathrm{e}^{ax}\cos^{n-2}bx\mathrm{d}x$

■ **对数**

1. $\int \ln x \, dx = x \ln x - x + C$

2. $\int \frac{dx}{x \ln x} = \ln |\ln x| + C$

3. $\int x^n \ln x \, dx = \frac{1}{n+1} x^{n+1} \left(\ln x - \frac{1}{n+1} \right) + C$

4. $\int (\ln x)^n dx = x(\ln x)^n - n \int (\ln x)^{n-1} dx$

5. $\int x^m (\ln x)^n dx = \frac{1}{m+1} x^{m+1} (\ln x)^n - \frac{n}{m+1} \int x^m (\ln x)^{n-1} dx$

■ **三角函数**

1. $\int \sin x \, dx = -\cos x + C$

2. $\int \cos x \, dx = \sin x + C$

3. $\int \tan x \, dx = -\ln |\cos x| + C$

4. $\int \cot x \, dx = \ln |\sin x| + C$

5. $\int \sec x \, dx = \ln \left| \tan \left(\frac{\pi}{4} + \frac{x}{2} \right) \right| + C = \ln |\sec x + \tan x| + C$

6. $\int \csc x \, dx = \ln \left| \tan \frac{x}{2} \right| + C = \ln |\csc x - \cot x| + C$

7. $\int \sec^2 x \, dx = \tan x + C$

8. $\int \csc^2 x \, dx = -\cot x + C$

9. $\int \sec x \tan x \, dx = \sec x + C$

10. $\int \csc x \cot x \, dx = -\csc x + C$

11. $\int \sin^2 x \, dx = \frac{x}{2} - \frac{1}{4} \sin 2x + C$

12. $\int \cos^2 x \, dx = \frac{x}{2} + \frac{1}{4} \sin 2x + C$

13. $\int \sin^n x \, dx = -\frac{1}{n} \sin^{n-1} x \cos x + \frac{n-1}{n} \int \sin^{n-2} x \, dx$

14. $\int \cos^n x \, dx = \frac{1}{n} \cos^{n-1} x \sin x + \frac{n-1}{n} \int \cos^{n-2} x \, dx$

15. $\int \frac{dx}{\sin^n x} = -\frac{1}{n-1} \frac{\cos x}{\sin^{n-1} x} + \frac{n-2}{n-1} \int \frac{dx}{\sin^{n-2} x}$

16. $\int \frac{dx}{\cos^n x} = \frac{1}{n-1} \frac{\sin x}{\cos^{n-1} x} + \frac{n-2}{n-1} \int \frac{dx}{\cos^{n-2} x}$

17.

$$\int \cos^m x \sin^n x \, dx$$
$$= \frac{1}{m+n} \cos^{m-1} x \sin^{n+1} x + \frac{m-1}{m+n} \int \cos^{m-2} x \sin^n x \, dx$$
$$= -\frac{1}{m+n} \cos^{m+1} x \sin^{n-1} x + \frac{n-1}{m+1} \int \cos^m x \sin^{n-2} x \, dx$$

18. $\int \sin ax \cos bx \, dx = -\frac{1}{2(a+b)} \cos(a+b)x - \frac{1}{2(a-b)} \cos(a-b)x + C$

19. $\int \sin ax \sin bx \, dx = -\frac{1}{2(a+b)} \sin(a+b)x + \frac{1}{2(a-b)} \sin(a-b)x + C$

20. $\int \cos ax \cos bx \, dx = \frac{1}{2(a+b)} \sin(a+b)x + \frac{1}{2(a-b)} \sin(a-b)x + C$

21. $\int \frac{dx}{a+b\sin x} = \begin{cases} \frac{2}{\sqrt{a^2-b^2}} \arctan \frac{a \tan \frac{x}{2}+b}{\sqrt{a^2-b^2}} + C & (a^2 > b^2) \\ \frac{1}{\sqrt{b^2-a^2}} \ln \left| \frac{a \tan \frac{x}{2}+b-\sqrt{b^2-a^2}}{a \tan \frac{x}{2}+b+\sqrt{b^2-a^2}} \right| + C & (a^2 < b^2) \end{cases}$

22. $\int \frac{dx}{a+b\cos x} = \begin{cases} \frac{2}{a+b} \sqrt{\frac{a+b}{a-b}} \arctan \left(\sqrt{\frac{a-b}{a+b}} \tan \frac{x}{2} \right) + C & (a^2 > b^2) \\ \frac{1}{a+b} \sqrt{\frac{a+b}{a-b}} \ln \left| \frac{\tan \frac{x}{2}+\sqrt{\frac{a+b}{b-a}}}{\tan \frac{x}{2}-\sqrt{\frac{a+b}{b-a}}} \right| + C & (a^2 < b^2) \end{cases}$

23. $\int \frac{dx}{a^2 \cos^2 x + b^2 \sin^2 x} = \frac{1}{ab} \arctan \left(\frac{b}{a} \tan x \right) + C$

24. $\int \frac{dx}{a^2 \cos^2 x - b^2 \sin^2 x} = \frac{1}{2ab} \ln \left| \frac{b \tan x + a}{b \tan x - a} \right| + C$

25. $\int x \sin ax \, dx = \frac{1}{a^2} \sin ax - \frac{1}{a} x \cos ax + C$

26. $\int x^2 \sin ax \, dx = -\frac{1}{a} x^2 \cos ax + \frac{2}{a^2} x \sin ax + \frac{2}{a^3} \cos ax + C$

27. $\int x \cos ax \, dx = \frac{1}{a^2} \cos ax + \frac{1}{a} x \sin ax + C$

28. $\int x^2 \cos ax \, dx = \frac{1}{a} x^2 \sin ax + \frac{2}{a^2} x \cos ax - \frac{2}{a^3} \sin ax + C$

■ **反三角函数** $(a > 0)$

1. $\int \arcsin \frac{x}{a} \, dx = x \arcsin \frac{x}{a} + \sqrt{a^2 - x^2} + C$

2. $\int x \arcsin \frac{x}{a} \, dx = \left(\frac{x^2}{2} - \frac{a^2}{4} \right) \arcsin \frac{x}{a} + \frac{x}{4} \sqrt{x^2 - x^2} + C$

3. $\int x^2 \arcsin \frac{x}{a} \, dx = \frac{x^3}{3} \arcsin \frac{x}{a} + \frac{1}{9} (x^2 + 2a^2) \sqrt{a^2 - x^2} + C$

4. $\int \arccos \frac{x}{a} \, dx = x \arccos \frac{x}{a} - \sqrt{a^2 - x^2} + C$

5. $\int x \arccos \frac{x}{a} \, dx = \left(\frac{x^2}{2} - \frac{a^2}{4} \right) \arccos \frac{x}{a} - \frac{x}{4} \sqrt{a^2 - x^2} + C$

6. $\int x^2 \arccos \frac{x}{a} \, dx = \frac{x^3}{3} \arccos \frac{x}{a} - \frac{1}{9} (x^2 + 2a^2) \sqrt{a^2 - x^2} + C$

7. $\int \arctan \frac{x}{a} \, dx = x \arctan \frac{x}{a} - \frac{a}{2} \ln(a^2 + x^2) + C$

8. $\int x \arctan \frac{x}{a} \, dx = \frac{1}{2} (a^2 + x^2) \arctan \frac{x}{a} - \frac{a}{2} x + C$

9. $\int x^2 \arctan \frac{x}{a} \, dx = \frac{x^3}{3} \arctan \frac{x}{a} - \frac{a}{6} x^2 + \frac{a^3}{6} \ln(a^2 + x^2) + C$

9.1.6 其它

■ **克拉夫特不等式** 若二叉树有 n 个叶子，深度分别为 $d_1, d_2, ..., d_n$，则 $\sum_{i=1}^{n} 2^{-d_i} \leq 1$，

当且仅当叶子都有兄弟时取等

9.2 几何公式

9.2.1 平面几何

■ **三角形的长度**

中线 $m_a = \sqrt{\frac{1}{2}b^2 + \frac{1}{2}c^2 - \frac{1}{4}a^2}$

高线长 $h_a = \frac{2\sqrt{s(s-a)(s-b)(s-c)}}{a}$

角平分线 $t_a = \frac{1}{b+c}\sqrt{(b+c+a)(b+c-a)bc}$

外接圆半径 $R = \frac{abc}{\sqrt{(a+b+c)(b+c-a)(a+c-b)(a+b-c)}}$

内切圆半径 $r = \frac{\sqrt{(a+b+c)(b+c-a)(a+c-b)(a+b-c)}}{2(a+b+c)}$

■ **三角形的面积**

$S = \frac{1}{2}ab\sin C = \frac{a^2 \sin B \sin C}{2\sin(B+C)} = \sqrt{p(p-a)(p-b)(p-c)} = \frac{1}{2}\begin{Vmatrix} a_x & a_y & 1 \\ b_x & b_y & 1 \\ c_x & c_y & 1 \end{Vmatrix}$，其中 $p = \frac{a+b+c}{2}$

■ **三角形奔驰定理**

P 为 $\triangle ABC$ 中一点，且 $S_{\triangle PBC} \cdot \overrightarrow{PA} + S_{\triangle PAC} \cdot \overrightarrow{PB} + S_{\triangle PAB} \cdot \overrightarrow{PC} = \vec{0}$

■ **托勒密定理**

狭义：凸四边形四点共圆当且仅当其两对对边乘积的和等于两条对角线的乘积

广义：四边形 $ABCD$ 两条对角线长分别为 m, n，则 $m^2n^2 = a^2c^2 + b^2d^2 - 2abcd\cos(A+C)$

■ **椭圆面积** $S = \pi ab$

■ **弧微分** $ds = \sqrt{[x'(t)]^2 + [y'(t)]^2}dt = \sqrt{1 + [f'(x)]^2}dx = \sqrt{r^2(\theta) + [r'(\theta)]^2}d\theta$

■ **费马点** 三角形费马点是指与三顶点距离之和最小的点。当有一个内角不小于 120° 时，费马点为此角对应顶点；当三角形的内角都小于 120° 时，据三角形各边向外做正三角形，连接新产生的三点与各自在原三角形中所对顶点，则三线交于费马点。

9.2.2 立体几何

■ **凸多面体欧拉公式** 对任意凸多面体，点、边、面数分别为 V, E, F，则 $V - E + F = 2$

■ **台体体积** $V = \frac{1}{3}h(S_1 + \sqrt{S_1 S_2} + S_2)$

■ **椭球体积** $V = \frac{4}{3}\pi abc$（都是半轴）

■ **四面体体积**

$V = \frac{1}{6}\begin{Vmatrix} p_x & p_y & p_z \\ q_x & q_y & q_z \\ r_x & r_y & r_z \end{Vmatrix}$，其中 $\vec{p} = \overrightarrow{OA}, \vec{q} = \overrightarrow{OB}, \vec{r} = \overrightarrow{OC}$;

$(12V)^2 = a^2d^2(b^2 + c^2 + e^2 + f^2 - a^2 - d^2) + b^2e^2(c^2 + a^2 + f^2 + d^2 - b^2 - e^2) + c^2f^2(a^2 + b^2 + d^2 + e^2 - c^2 - f^2) - a^2b^2c^2 - a^2e^2f^2 - d^2b^2f^2 - d^2e^2c^2$，其中 $a = AB, b = BC, c = CA, d = OC, e = OA, f = OB$

■ **旋转体（一、二象限，绕 x 轴）**

体积 $V = \pi \int_a^b f^2(x)dx$

侧面积 $F = 2\pi \int f(x)ds = 2\pi \int_a^b \sqrt{1 + [f'(x)]^2}dx$

（空心）质心

$$X = \frac{1}{M}\int_\alpha^\beta x(t)\rho(t)\sqrt{[x'(t)]^2 + [y'(t)]^2}dt$$

$$Y = \frac{1}{M}\int_\alpha^\beta y(t)\rho(t)\sqrt{[x'(t)]^2 + [y'(t)]^2}dt$$

（空心）转动惯量

$$J_x = \int_\alpha^\beta y^2(t)\rho(t)\sqrt{[x'(t)]^2 + [y'(t)]^2}dt$$

$$J_y = \int_\alpha^\beta x^2(t)\rho(t)\sqrt{[x'(t)]^2 + [y'(t)]^2}dt$$

古鲁丁定理：平面上一条质量分布均匀曲线绕一条不通过它的直线轴旋转一周，所得到的旋转体之侧面积等于它的质心绕同一轴旋转所得圆的周长乘以曲线的弧长。

9.3 经典博弈

■ **Nim 博弈**

问题：n 堆石子，每次取一堆中 x 个 $(x > 0)$，取完则胜。

奇异态（后手胜）：$a_1 \ xor \ a_2 \ xor \ ... \ xor \ a_n = 0$

■ **Bash 博弈**

问题：n 个石子，每次 x 个 $(0 < x \leq m)$，取完则胜。

奇异态（后手胜）：$n \equiv 0 \pmod{(m+1)}$

■ Wythoff 博弈

问题：2 堆石子分别 x, y 个 $(x > y)$，每次取一堆中 x 个 $(x > 0)$，或两堆中分别 x 个 $(x > 0)$，取完则胜。
奇异态（后手胜）：$\left\lfloor \frac{\sqrt{5}+1}{2}(x-y) \right\rfloor = y$

■ Fibonacci 博弈

问题：n 个石子，先手第一次取 x 个 $(0 < x < n)$，之后每次取 x 个 $(0 < x \le$ 上一次取数的两倍)，取完则胜。
奇异态（**先**手胜）：n 不是斐波那契数

9.4 部分质数

100003, 200003, 300007, 400009, 500009, 600011, 700001, 800011, 900001,
1000003, 2000003, 3000017, 4100011, 5000011, 8000009, 9000011,
10000019, 20000003, 50000017, 50100007,
100000007, 100200011, 200100007, 250000019

10 语法

精选部分函数，无特别说明则为 98 标准

10.1 C

10.1.1 <cstdio>

```
// ─────────────────────开关文件（流）─────────────────────
FILE * fopen ( const char * filename, const char * mode );
FILE * freopen ( const char * filename, const char * mode, FILE * stream );
int fclose ( FILE * stream );
// 1.  fopen是载入流；freopen是流的重定向，将filename的文件载至stream
// 2.  mode可选"r"(read),"w"(write),"a"(append)...后可加"b"(binary),"+"(
//     update)或"b+"
// ─────────────────────读写流─────────────────────
int printf ( const char * format, ... );
int scanf ( const char * format, ... );
int fprintf ( FILE * stream, const char * format, ... );
int fscanf ( FILE * stream, const char * format, ... );
int sprintf ( char * str, const char * format, ... );
int sscanf ( const char * s, const char * format, ...);
size_t fread ( void * ptr, size_t size, size_t count, FILE * stream );
// 1.  f~针对文件，s~针对cstring
// 2.  返回成功读入（输出）元素个数
// 3.  判断读入末尾：while (~scanf()), while (scanf()!=EOF)
// 4.  scanf一个元素[前]，忽略满足<cctype>isspace()的字符，注意读元素[后]的未
//     读入！
// 5.  scanf：%[*][width][length]specifier. *表读指定类型但不保存，width表读
//     入最大字符数；%[ABC]仅读ABC三种字符，%[A-Z]只读大写字母，%[^ABC]表过滤ABC
// 6.  printf：%[flags][width][.precision][length]specifier.
//     [flags]：-左对齐；+数字符号强制显示；0数前补0至列宽；(空格)正数前加空
//     格负数前加负号；#类型o/x/X前加0/0x/0X，类型e/E/f/g/G强制输出小数点，类型g/
//     G保留尾部0
//     specifier：d有符号十进制整；u无符号10进制整；o无符号8进制整；x/X无符号
//     十六进制整（小/大写）；e/E科学计数法double（e小/大写）
// 7.  printf时，百分号%%，单引号\'，双引号\"，反斜杠\\
// 8.  输入特别难搞时，开大小为bufsize的数组buf，然后fread(buf,1,bufsize,
//     stdin)
// ─────────────────────逐字符读写─────────────────────
int getc ( FILE * stream );
int getchar ( void );
char * gets ( char * str );
int putc ( int character, FILE * stream );
int putchar ( int character );
int puts ( const char * str );
int ungetc ( int character, FILE * stream );
// 1.  ungetc退回字符到输入流中
// 2.  getchar读进'\n'，gets不读进'\n'
// 3.  注意以上对于文件末尾int返回值为EOF而非0
```

10.1.2 <cctype>

```
int toupper ( int c );
int tolower ( int c );
int is~ ( int c );
//isspace 空格' '，TAB'\t'，换行'\n'，回车'\r'，'\v'，'\f'
//isupper大写字母，islower小写字母，isalpha字母，isdigit数字
```

10.1.3 <cstring>

```
// ─────────────────────修改─────────────────────
void * memset ( void * ptr, int value, size_t num );
void * memcpy ( void * destination, const void * source, size_t num );
char * strcpy ( char * destination, const char * source );
char * strncpy ( char * destination, const char * source, size_t num );
char * strcat ( char * destination, const char * source );
char * strncat ( char * destination, const char * source, size_t num );
// 1.  以上在后面一定有'\0'的有strcpy, strcat和strncat，注意strncpy不自动加！
// ─────────────────────比较─────────────────────
int memcmp ( const void * ptr1, const void * ptr2, size_t num );
int strcmp ( const char * str1, const char * str2 );
int strncmp ( const char * str1, const char * str2, size_t num );
// ─────────────────────查找─────────────────────
const void * memchr ( const void * ptr, int value, size_t num );
      void * memchr (       void * ptr, int value, size_t num );
const char * strchr ( const char * str, int character );
      char * strchr (       char * str, int character );
const char * strrchr ( const char * str, int character );
      char * strrchr (       char * str, int character );
const char * strstr ( const char * str1, const char * str2 );
      char * strstr (       char * str1, const char * str2 );
```

```
22  size_t strspn ( const char * str1, const char * str2 );
23  size_t strcspn ( const char * str1, const char * str2 );
24  const char * strpbrk ( const char * str1, const char * str2 );
25        char * strpbrk (       char * str1, const char * str2 );
26  char * strtok ( char * str, const char * delimiters );
27  // 1.  strrchr的搜索包括\0，所以strrchr(s,0)返回末尾指针
28  // 2.  strspn返回str1开头最长连续多少个字符都在str2中出现，strcspn相反意义
29  // 3.  strpbrk返回str1中最先出现在str2中的字符的指针
30  // 4.  strtok通过delimiters字符集分割str（不包含那些字符），每次取一个分割出
         的子串用p=strtok(NULL,delimiters)，直到p为NULL
31  //——————————————————————其它——————————————————————
32  size_t strlen ( const char * str );
```

11 经典错误

= 和 == 混淆；scanf 没加 &；爆数组/数据范围