Contents

1 基础/配置/编译

1.1 一般母版

```cpp
#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*10+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 fread

```cpp
char buf[buff];
char *bs=0,*bt=0;
void iobegin()
{
  int l=fread(buf,1,buff,stdin);
  bs=buf; bt=bs+l;
}
```

```
char get()  //替换getchar
{
    if (bs==bt) iobegin();
    return *(bs++);
}
```

1.3 编译

```
//====================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);
```

1.4 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;
}
int snoob2(int x)   //g++
{
    int t=x|(x-1);
    return (t+1)|(((~t&-~t)-1)>>(__builtin_ctz(x)+1));
}
//======================按位反转======================
int reverse(int x)
{
    x=((x&0x55555555)<<1)|((x&0xAAAAAAAA)>>1);
    x=((x&0x33333333)<<2)|((x&0xCCCCCCCC)>>2);
    x=((x&0x0F0F0F0F)<<4)|((x&0xF0F0F0F0)>>4);
    x=((x&0x00FF00FF)<<8)|((x&0xFF00FF00)>>8);
    x=((x&0x0000FFFF)<<16)|((x&0xFFFF0000)>>16);
    return x;
}
//=======注意!!以下g++下才能用;ll则在函数名后加ll========
int __builtin_popcount(unsigned int x); //1的个数
int __builtin_clz(unsigned int x);      //前缀0的个数
//x为int时,31-__builtin_clz(x) 等价于 int(log(x)/log(2))
int __builtin_ctz(unsigned int x);      //后缀0的个数
int __builtin_parity(unsigned int x);   //1的个数%2
```

1.5 离散化

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

1.6 Linux 对拍

```
g++ $2 -o 1.out
g++ $3 -o 2.out
cnt=0;
while true; do
g++ $1 -o dm.out
./dm.out>dm.txt
./1.out<dm.txt>1.txt
./2.out<dm.txt>2.txt
if diff 1.txt 2.txt; then let "cnt+=1"; echo ${cnt};
else exit 0;
fi
done
```

1.7 vimrc

```
runtime! debian.vim

if has("syntax")
  syntax on
endif

if filereadable("/etc/vim/vimrc.local")
  source /etc/vim/vimrc.local
endif


colo torte
set nu
set ts=4
set sw=4
map <C-A> ggVG"+y
map <F2> :w<CR>
```

```
map <F3> :browse e<CR>
map <F4> :browse vsp<CR>
map <F5> :call Run()<CR>
func! Run()
  exec "w"
  exec "!g++ -Wall % -o %<"
  exec "!./%<"
endfunc
```

1.8 gdb

```
//g++ -g a.cpp -o a;gdb --args a 1
int main(int gdb)
{
  if (gdb>1)
  {
    freopen("in","r",stdin);
    freopen("out","w",stdout);
  }

  //...

  return 0;
}
```

2 数学

2.1 分数类

```
int gcd(int x,int y){return y?gcd(y,x%y):x;}
struct frac
{
  int x,y;  //符号放在x
  frac adjust()
  {
    if (!y) {x=0; return *this;}
    if (!x) {y=1; return *this;}
    int sg=(x>0?1:-1)*(y>0?1:-1);
    int t=gcd(x=abs(x),y=abs(y)); x=x/t*sg; y/=t;
    return *this;
  }
  frac(){}
  frac(int a,int b){x=a;y=b;this->adjust();}
  frac(char *a,bool improp=0) //improp假分数
  {
        int t,sg=1;
        if (*a=='-') {sg=-1; a++;}
        if (improp&&strchr(a,' ')) {sscanf(a,"%d %d/%d",&t,&x,&y); x+=t*y;}
        else if (strchr(a,'/')) sscanf(a,"%d/%d",&x,&y);
        else {sscanf(a,"%d",&x); y=1;}
        x*=sg; this->adjust();
  }
  char* c_str(bool improp=0)
  {
    char *res=new char[50](),t[50];
    this->adjust();
    if (x==0) {res[0]='0'; return res;}
    if (x<0) {strcat(res,"-"); x=-x;}
    if (improp&&x/y&&x%y){sprintf(t,"%d ",x/y); strcat(res,t); x%=y;}
    sprintf(t,"%d",x); strcat(res,t);
    if (y!=1) {sprintf(t,"/%d",y); strcat(res,t);}
    return res;
  }
};
bool operator==(frac a,frac b){a.adjust();b.adjust();return a.x==b.x&&a.y==b.y;}
bool operator!=(frac a,frac b){a.adjust();b.adjust();return !(a.x==b.x&&a.y==b.y);}
bool 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;}
bool 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;}
frac operator+(frac a,frac b){return frac(a.x*b.y+a.y*b.x,a.y*b.y).adjust();}
frac operator-(frac a,frac b){return frac(a.x*b.y-a.y*b.x,a.y*b.y).adjust();}
frac operator*(frac a,frac b){return frac(a.x*b.x,a.y*b.y).adjust();}
frac operator/(frac a,frac b){return frac(a.x*b.y,a.y*b.x).adjust();}
const frac nonfrac=frac(0,0);
const frac zerofrac=frac(0,1);
```

2.2 高精度类

```
//要sqrt就一定要len和dcm是偶数
//不可以出现如big x=y;的东西，必须分开成big x;x=y;
#define len 3000
#define dcm 2000
struct big
{
    int _[len+2];

    int& operator[](int x){return _[x];}
    big(){memset(_,0,sizeof(int)*(len+2));}
    big(char*x)
    {
        memset(_,0,sizeof(int)*(len+2));
        char *y=x+strlen(x)-1,*z=strchr(x,'.'),*i;
        if (!z) z=y+1;
        int t=dcm-(z-x);
        tr(i,x,y) if(i!=z&&t>=1&&t<=len) _[++t]=*i-'0';
    }

    big& operator=(const big&x){memcpy(_,x._,sizeof(int)*(len+2));return *this;}
    char* c_str()
    {
        char *s=new char[len]; int l,r,i=0,k;
```

```
        tr(l,1,len) if(_[l]>0||l==dcm) break;
        rtr(r,len,1) if(_[r]>0||r==dcm) break;
        tr(k,l,r){if(k==dcm+1)s[i++]='.';s[i++]=_[k]+'0';}
        s[i]=0; return s;
    }
};
void carry(int*x,int y){*(x-1)+=((*x+=y)+10000)/10-1000;*x=(*x+10000)%10;}
int comp(big x,big y) //O(len)
{
    int i;
    tr(i,1,len) if (x[i]!=y[i]) break;
    return i>len?0:(x[i]>y[i]?1:-1);
}
big operator+(big x,big y) //O(len)
{
    big z; int i;
    rtr(i,len,1) carry(&z[i],x[i]+y[i]);
    return z;
}
big operator-(big x,big y) //O(len)
{
    big z; int i;
    rtr(i,len,1) carry(&z[i],x[i]-y[i]);
    return z;
}
big operator*(big x,big y) //O(len^2)
{
    big z; int i,j;
    rtr(i,len,1) rtr(j,min(dcm+len-i,len),max(dcm+1-i,1))
        carry(&z[i+j-dcm],x[i]*y[j]);
    return z;
}
big operator/(big x,big y) //O(len^2)
{
    big z,t,tmp[10]; int i,j,k;
    tr(k,1,9) tmp[k]=tmp[k-1]+y;
    tr(j,1,len-dcm) t[j+dcm]=x[j];
    j--;
    tr(i,1,len)
    {
        tr(k,1,len-1) t[k]=t[k+1];
        t[len]=++j<=len?x[j]:0;
        tr(k,1,9) if (comp(tmp[k],t)>0) break;
        z[i]=--k;
        t=t-tmp[k];
    }
    return z;
}
big operator+(big x,int y)  //only dcm=len
{
    big z; int i;
    carry(&z[len],x[len]+y);
    rtr(i,len-1,1) carry(&z[i],x[i]);
    return z;
}
big operator-(big x,int y)  //only dcm=len
{
    big z; int i;
    carry(&z[len],x[len]-y);
    rtr(i,len-1,1) carry(&z[i],x[i]);
    return z;
}
big operator*(big x,int y)  //only dcm=len
{
    big z; int i;
    carry(&z[len],x[len]*y);
    rtr(i,len-1,1) carry(&z[i],x[i]*y);
    return z;
}
pair<big,int> operator/(big x,int y)  //only dcm=len
{
  big z; int d=0,i;
  tr(i,1,len)
    {
      z[i]=(d*10+x[i])/y;
      d=(d*10+x[i])%y;
    }
    return make_pair(z,d);
}
int sqrt_deal(big&y,int a,int b,int l)
{
    int t=a+y[b]%10-9;
    if(2*b>l)t-=(y[2*b-l])/10;
    if (b>=0&&!(a=sqrt_deal(y,t/10,b-1,l))) y[b]+=(t+999)%10-y[b]%10;
    return a;
}
big sqrt(big x) //O(len^2)
{
    int l,t=dcm/2; big y,z; y=x;
    for(l=1;l<=len;l++)
    {
        y[++l]+=10;
        while (!sqrt_deal(y,0,l,l)) y[l]+=20;
        z[++t]=y[l]/20; y[l]-=10;
    }
    return z;
}
big floor(big x)
{
    big z; z=x; int i;
    tr(i,dcm+1,len) z[i]=0;
    return z;
}
big ceil(big x){return comp(x,floor(x))==0?x:floor(x+big("1"));}
```

2.3 矩阵类

```
//^是O(n^2.8)的矩阵乘法；一定要用引用传递mat，不然会爆
#define maxn 130
struct mat
```

```
{
    int n,m,a[maxn][maxn];
    mat(int nn=0,int mm=0):n(nn),m(mm){}
    int* operator[](int x){return a[x];}
};
const mat nonmat(-1,-1);
mat unit(int n)
{
    mat res(n,n);
    int i,j;
    tr(i,1,res.n) tr(j,1,res.n) if (i==j) res[i][j]=1; else res[i][j]=0;
    return res;
}
mat operator+(mat&a,mat&b)
{
    if (a.n!=b.n||a.m!=b.m) return nonmat;
    mat c(a.n,a.m); int i,j;
    tr(i,1,a.n) tr(j,1,a.m) c[i][j]=a[i][j]+b[i][j];
    return c;
}
mat operator-(mat&a,mat&b)
{
    if (a.n!=b.n||a.m!=b.m) return nonmat;
    mat c(a.n,a.m); int i,j;
    tr(i,1,a.n) tr(j,1,a.m) c[i][j]=a[i][j]-b[i][j];
    return c;
}
mat operator*(mat&a,mat&b)
{
    if (a.m!=b.n) return nonmat;
    mat c(a.n,b.m); int i,j,k;
    tr(i,1,a.n) tr(j,1,b.m)
    {
        c[i][j]=0;
        tr(k,1,a.m) c[i][j]+=a[i][k]*b[k][j];
    }
    return c;
}
mat operator^(mat&a,ll b)
{
  if (a.n!=a.m) return nonmat;
    mat res=unit(a.n);
    while (b)
    {
        if (b&1) res=res*a;
        a=a*a;
        b>>=1;
    }
    return res;
}
//=================================================================
void _as(mat&a,int x1,int y1,mat&b,int x2,int y2,int nn,int mm,bool chnm=0) //
    assign
{
    int i,j; tr(i,1,nn) tr(j,1,mm) a[x1+i-1][y1+j-1]=b[x2+i-1][y2+j-1];
    if (chnm) {a.n=x1+nn-1; a.m=y1+mm-1;}
}
void _st(mat&a,mat&b,mat&c,int n,int m,int k)  //strassen
{
    if (n<=32||m<=32||k<=32){c=a*b;return;}
    c.n=n; c.m=m;
    n>>=1; m>>=1; k>>=1;
    mat a11,a12,a21,a22,b11,b12,b21,b22,m1,m2,m3,m4,m5,m6,m7,t1,t2;
    _as(a11,1,1,a,1,1,n,k,1);       _as(a12,1,1,a,1,k+1,n,k,1);
    _as(a21,1,1,a,n+1,1,n,k,1);     _as(a22,1,1,a,n+1,k+1,n,k,1);
    _as(b11,1,1,b,1,1,k,m,1);       _as(b12,1,1,b,1,m+1,k,m,1);
    _as(b21,1,1,b,k+1,1,k,m,1);     _as(b22,1,1,b,k+1,m+1,k,m,1);
    t1=a11+a22;t2=b11+b22;          _st(t1,t2,m1,n,m,k);
    t1=a21+a22;t2=b11;              _st(t1,t2,m2,n,m,k);
    t1=a11;t2=b12-b22;              _st(t1,t2,m3,n,m,k);
    t1=a22;t2=b21-b11;              _st(t1,t2,m4,n,m,k);
    t1=a11+a12;t2=b22;              _st(t1,t2,m5,n,m,k);
    t1=a21-a11;t2=b11+b12;          _st(t1,t2,m6,n,m,k);
    t1=a12-a22;t2=b21+b22;          _st(t1,t2,m7,n,m,k);
    t1=m1+m4; t1=t1-m5; t1=t1+m7;   _as(c,1,1,t1,1,1,n,m);
    t1=m3+m5;                       _as(c,1,m+1,t1,1,1,n,m);
    t1=m2+m4;                       _as(c,n+1,1,t1,1,1,n,m);
    t1=m1-m2; t1=t1+m3; t1=t1+m6;   _as(c,n+1,m+1,t1,1,1,n,m);
}
int __enlarge(int x){int t=1<<(31-__builtin_clz(x)); return t==x?x:(t<<1);}
mat operator^(mat&a,mat&b)
{
    if (a.m!=b.n) return mat(-1,-1);
    int n=__enlarge(a.n),m=__enlarge(b.m),k=__enlarge(a.m);
    mat c; _st(a,b,c,n,m,k);
    c.n=a.n; c.m=b.m;
    return c;
}
```

2.4 GCD

$O(logN)$

```
int __gcd(int x,int y)  //<algorithm>且g++才能用
int gcd(int x,int y){return y?gcd(y,x%y):x;}
int gcd(int x,int y){while(y){int z=x%y; x=y; y=z;}return x;}
```

2.5 快速乘法

$O(logN)$

```
//快速乘法
ll mul(ll x,ll y,ll mod)
{
  ll res=0;
  for(;y;y>>=1)
  {
    if (y&1) res=(res+x)%mod;
    x=(x+x)%mod;
  }
```

```
	return res;
}

//int128法
ll mul(__int128 x,__int128 y,__int128 mod)   //同理存在__float128
{
    return x*y%mod;
}

//汇编法
ll mul(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;
}
```

2.6 快速幂

$O(logM)$

```
int pow(int x,int y,int mod)
{
    int res=1;
    while(y)
    {
        if (y&1) res=1LL*res*x%mod;
        x=1LL*x*x%mod;
        y>>=1;
    }
    return res;
}
```

2.7 找数列线性递推式

```
int nn,n,m,i,j,b[maxn];
double a[maxn][maxn],ans[maxn];
bool gauss()
{
    int i=1,j,k,x,y,arb=m;
    double t;
    tr(j,1,m)
    {
        tr(k,i,n) if (cmp(a[k][j])!=0) break;
        if (k>n) continue;
        arb--;
        if (k!=i) tr(y,j,m+1) swap(a[i][y],a[k][y]);
        tr(x,1,n) if(x!=i)
        {
            t=a[x][j]/a[i][j];
            tr(y,j,m+1) a[x][y]=a[x][y]-a[i][y]*t;
        }
        if (++i>n) break;
    }
    //-------------
    tr(i,1,n)
    {
        tr(j,1,m) if (cmp(a[i][j])!=0) break;
        if (j>m&&cmp(a[i][j])!=0) return 0;
    }
    if (arb) return 0;
    //-------------
    rtr(i,m,1)
    {
        ans[i]=a[i][m+1];
        rtr(j,m,i+1) ans[i]=ans[i]-ans[j]*a[i][j];
        ans[i]=ans[i]/a[i][i];
    }
    return 1;
}
int main()
{
  scanf("%d",&nn);
  tr(i,1,nn) scanf("%d",&b[i]);
  tr(m,1,(nn+1)/2)
  {
    n=0;
    tr(i,m,nn)
    {
      n++;
      tr(j,1,m-1) a[n][j]=b[i-j];
      a[n][m]=1; a[n][m+1]=b[i];
    }
    if (gauss()) break;
  }
  printf("a[n] = ");
  tr(i,1,m-1) printf("%+lf a[n-%d] ",ans[i],i);
  printf("%+lf\n",ans[m]);
  return 0;
}
```

2.8 筛素数-欧拉筛法

$O(N)$

```
int prime[maxm],a[n];
bool pprime[n];
void EulerPrime()
{
  int i,j;
  tr(i,2,n) pprime[i]=1;
  tr(i,2,n)
  {
    if (pprime[i]) prime[++m]=i;
```

```
    tr(j,1,m)
    {
      if (i*prime[j]>n) break;
      pprime[i*prime[j]]=0;
      if (i%prime[j]==0) break;
    }
  }
}
```

2.9 线性方程组-高斯消元

$O(N^3)$

```
//待测
int n,m;
frac a[maxn][maxm],ans[maxn];
void gauss()
{
    int i=1,j,k,x,y,arb=m;
    frac t;
    tr(j,1,m)
    {
        tr(k,i,n) if (a[k][j]!=zerofrac) break;
        if (k>n) continue;
        arb--;
        if (k!=i) tr(y,j,m+1) swap(a[i][y],a[k][y]);
        tr(x,1,n) if(x!=i)
        {
            t=a[x][j]/a[i][j];
            tr(y,j,m+1) a[x][y]=a[x][y]-a[i][y]*t;
        }
        if (++i>n) break;
    }
    //-------------
    tr(i,1,n)
    {
        tr(j,1,m) if (a[i][j]!=zerofrac) break;
        if (j>m&&a[i][j]!=zerofrac) {printf("No Solution.\n"); return;}
    }
    if (arb) {printf("Arbitrary constants: %d\n",arb); return;}
    //-------------
    rtr(i,m,1)
    {
        ans[i]=a[i][m+1];
        rtr(j,m,i+1) ans[i]=ans[i]-ans[j]*a[i][j];
        ans[i]=ans[i]/a[i][i];
    }
    tr(i,1,m) printf("x[%d] = %s\n",i,ans[i].c_str());
}
```

2.10 高阶代数方程求根-求导

$O(N^3 * S)$，S 取决于精度

```
//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
double a[maxn][maxn],ans[maxn][maxn];
int n,anss[maxn];
double get(int x,double y)
{
    int i; double res=0;
    rtr(i,x,0) res=res*y+a[x][i];
    return res;
}
void dich(int x,double ll,double rr)
{
    if (cmp(get(x,ll))==0){ans[x][++anss[x]]=ll;return;}
    if (cmp(get(x,rr))==0){ans[x][++anss[x]]=rr;return;}
    if (cmp(get(x,ll)*get(x,rr))>0) return;
    double l=ll,r=rr,mid;
    while (l+eps<r) //亦可改为循环一定次数
    {
        int tl=cmp(get(x,l)),tm=cmp(get(x,mid=(l+r)/2));
        if (tl==0) break;
        if (tl*tm>=0) l=mid; else r=mid;
    }
    ans[x][++anss[x]]=l;
}
void work()
{
    int i,j; double l,r;
    rtr(i,n-1,1) tr(j,0,i) a[i][j]=a[i+1][j+1]*(j+1);
    tr(i,0,n-1)
    {
        l=-oo;
        tr(j,1,anss[i]){dich(i+1,l,r=ans[i][j]); l=r;}
        dich(i+1,l,oo);
    }
    tr(i,1,anss[n]) printf("%.10lf\n",ans[n][i]);
}
```

2.11 FFT

$O(NlogN)$

```
#define pi acos(-1.0)
#define L 19
const int N=1<<L;
typedef complex<double> C;
int n,m,i;
int R[N],c[N];
C a[N],b[N];
void fft(C *a,int n,int f)
{
  int i,j,k;
  tr(i,0,n-1) if (i<R[i]) swap(a[i],a[R[i]]);
  for(i=1;i<n;i<<=1)
  {
    C wn(cos(pi/i),f*sin(pi/i));
```

```
    for(j=0;j<n;j+=(i<<1))
    {
      C w(1,0);
      tr(k,0,i-1)
      {
        C x=a[j+k],y=w*a[j+k+i];
        a[j+k]=x+y; a[j+k+i]=x-y;
        w*=wn;
      }
    }
  }
  if(f==-1) tr(i,0,n-1) a[i]/=n;
}
void mul(int na,C *a,int nb,C *b,int &nc,int *c)  //c=a=a*b
{
  int len=0,n;
  nc=na+nb;
  for(n=1;n<=nc+2;n<<=1) len++;
  tr(i,na+1,n-1) a[i]=0;
  tr(i,nb+1,n-1) b[i]=0;
  tr(i,0,n-1) R[i]=(R[i>>1]>>1)|((i&1)<<(len-1));
  fft(a,n,1); fft(b,n,1);
  tr(i,0,n) a[i]*=b[i];
  fft(a,n,-1);
  tr(i,0,nc) c[i]=int(a[i].real()+0.1);
}
void init1()  //大数乘法
{
  static char sa[N],sb[N];
  while (~scanf("%s%s",sa,sb))
  {
    memset(c,0,sizeof(c));
    n=strlen(sa)-1; m=strlen(sb)-1;
    tr(i,0,n) a[i]=sa[n-i]-'0';
    tr(i,0,m) b[i]=sb[m-i]-'0';
    mul(n,a,m,b,m,c);
    tr(i,0,m) if (c[i]>=10)
    {
      c[i+1]+=c[i]/10; c[i]%=10;
      if (i==m) m++;
    }
    while (!c[m]&&m) m--;
    rtr(i,m,0) printf("%d",c[i]); puts("");
  }
}
void init2()  //多项式乘法
{
  int x;
  while (~scanf("%d%d",&n,&m))
  {
    tr(i,0,n) scanf("%d",&x),a[i]=x;
    tr(i,0,m) scanf("%d",&x),b[i]=x;
    mul(n,a,m,b,m,c);
    tr(i,0,m) printf("%d ",c[i]);
    puts("");
  }
}
```

2.12 NTT

$O(NlogN)$

费马质数 $r \cdot 2^k+1$ 作模则 N 最大约为 2^k

$2281701377 = 17*2^{27}+1$(平方刚好不爆 LL), $1004535809 = 479*2^{21}+1, 998244353 = 119*2^{23}+1$，原根都是 3;

$786433 = 3*2^{18}+1$ 原根是 10，$880803841 = 105*2^{23}+1$ 原根是 26

```
const int P=998244353;
const int N=1<<19;
const int G=3;  //原根
int R[N],w[2][N];
void ntt(int *a,int n,int f)
{
  int i,j,k,t,l,x,y;
  tr(i,0,n-1) if (i<R[i]) swap(a[i],a[R[i]]);
  for(i=1;i<n;i<<=1)
    for(j=0,t=n/(i<<1);j<n;j+=(i<<1))
      for(k=0,l=0;k<i;k++,l+=t)
      {
        x=1LL*a[i+j+k]*w[f][l]%P;
        y=a[j+k];
        a[j+k]=(x+y)%P;
        a[i+j+k]=(y+P-x)%P;
      }
  if(f)
  {
    t=pow(n,P-2,P);
    tr(i,0,n-1) a[i]=1LL*a[i]*t%P;
  }
}
void mul(int na,int *a,int nb,int *b,int &nc) //a=a*b
{
  int len=0,i,n;
  nc=na+nb;
  for(n=1;n<=nc+2;n<<=1) len++;
  tr(i,na+1,n-1) a[i]=0;
  tr(i,nb+1,n-1) b[i]=0;
  tr(i,0,n-1) R[i]=(R[i>>1]>>1)|((i&1)<<(len-1));
  int v=pow(G,(P-1)/n,P);
  int dv=pow(v,P-2,P);
  w[0][0]=w[1][0]=1;
  tr(i,1,n-1)
  {
    w[0][i]=1LL*w[0][i-1]*v%P;
    w[1][i]=1LL*w[1][i-1]*dv%P;
  }
  ntt(a,n,0); ntt(b,n,0);
  tr(i,0,n) a[i]=(1LL*a[i]*b[i])%P;
  ntt(a,n,1);
}
```

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

```
#define maxpn 20//100010
#define nonx 1E100
#define eps 1E-8
const double pi=acos(-1.0);
//==========================================================
int cmp(double x)
{
    if (x>eps) return 1;
    if (x<-eps) return -1;
    return 0;
}
double sqr(double a){return a*a;}
int gcd(int a,int b){return a%b==0?b:gcd(b,a%b);}
//==========================================================
struct point
{
    double x,y;
    point(){}
    point(double a,double b){x=a;y=b;}

    friend point operator+(point a,point b){return point(a.x+b.x,a.y+b.y);}
    friend point operator-(point a,point b){return point(a.x-b.x,a.y-b.y);}
    friend point operator-(point a){return point(-a.x,-a.y);}
    friend double operator*(point a,point b){return a.x*b.x+a.y*b.y;}
    friend point operator*(double a,point b){return point(a*b.x,a*b.y);}
    friend point operator*(point a,double b){return point(a.x*b,a.y*b);}
    friend point operator/(point a,double b){return point(a.x/b,a.y/b);}
    friend double operator^(point a,point b){return a.x*b.y-a.y*b.x;}
    friend bool operator==(point a,point b){return cmp(a.x-b.x)==0&&cmp(a.y-b.y)==0;}
    friend bool operator!=(point a,point b){return cmp(a.x-b.x)!=0||cmp(a.y-b.y)!=0;}
} ;
const point nonp=point(nonx,nonx);
struct line
{
    point a,b;
    line(){}
    line(point pa,point pb){a=pa;b=pb;}
    point dir(){return b-a;}
} ;
const line nonl=line(nonp,nonp);
struct circle
{
    point o; double r;
    circle(){}
    circle(point a,double b){o=a;r=b;}
} ;
struct triangle//t 因triangle亦属polygon，故省去许多函数
{
    point a,b,c;
    triangle(){}
    triangle(point ta,point tb,point tc){a=ta;b=tb;c=tc;}
} ;
struct polygon
{
    int n; point a[maxpn];  //逆时针！
    polygon(){}
    polygon(triangle t){n=3;a[1]=t.a;a[2]=t.b;a[3]=t.c;}
    point& operator[](int _){return a[_];}
} ;
//==========================================================
double sqr(point a){return a*a;}
double len(point a){return sqrt(sqr(a));}    //模长
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
double angle(point a,point b){return acos(a*b/len(a)/len(b));}   //夹角
point reflect(point a,point b){return 2*a-b;}    //以a为中心对称
//==========================================================
point quad(double A,double B,double C)
{
    double delta=sqr(B)-4*A*C;
    if (delta<0) return nonp;
    return point((-B-sqrt(delta))/(2*A),(-B+sqrt(delta))/(2*A));
}
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
double dist(point a,line b){return ((a-b.a)^(b.b-b.a))/len(b.dir());}    //点到线距离
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
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
bool online(point a,line b){return cmp((a-b.a)^b.dir())==0;} //判断点在直线上
bool parallel(line a,line b){return cmp(a.dir()^b.dir())==0;}    //判断两线平行
point cross(line a,line b)   //线交
{
    double t;
    if (cmp(t=a.dir()^b.dir())==0) return nonp;
    return a.a+((b.a-a.a)^b.dir())/t*a.dir();
}
//==========================================================
double S(circle a){return pi*sqr(a.r);}  //面积
double C(circle a){return 2*pi*a.r;} //周长
line cross(line a,circle b)  //线圆交
{
    point t=quad(sqr(a.dir()),2*a.dir()*(a.a-b.o),sqr(a.a-b.o)-sqr(b.r));
    if (t==nonp) return nonl;
    return line(a.a+t.x*a.dir(),a.a+t.y*a.dir());
}
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
line cross(circle a,circle b)
{
```

```
    double d=len(a.o-b.o);
    if (cmp(abs(a.r-b.r)-d)>0||cmp(d-(a.r+b.r))>0) return nonl;
    double c=acos((sqr(a.r)+sqr(d)-sqr(b.r))/(2.0*a.r*d));
    point v=(b.o-a.o)/d*a.r;
    return line(a.o+rotate(v,-c),a.o+rotate(v,c));
}
//line tangent(point a,circle b){}
//pair<line,line> tangent(circle a,circle b){}
//double unionS(int n,circle*a) //圆面积并
//{}
//==========================================================
double S(triangle a){return abs((a.b-a.a)^(a.c-a.a))/2;} //面积
double C(triangle a){return len(a.a-a.b)+len(a.a-a.c)+len(a.a-a.c);} //周长
circle outcircle(triangle a) //外接圆
{
    circle res; point t1=a.b-a.a,t2=a.c-a.a;
    double t=2*t1^t2;
    res.o.x=a.a.x+(sqr(t1)*t2.y-sqr(t2)*t1.y)/t;
    res.o.y=a.a.y+(sqr(t2)*t1.x-sqr(t1)*t2.x)/t;
    res.r=len(res.o-a.a);
    return res;
}
circle incircle(triangle a)  //内切圆
{
    circle res; double x=len(a.b-a.c),y=len(a.c-a.a),z=len(a.a-a.b);
    res.o=(a.a*x+a.b*y+a.c*z)/(x+y+z);
    res.r=dist(res.o,line(a.a,a.b));
    return res;
}
point gc(triangle a){return (a.a+a.b+a.c)/3;}    //重心
point hc(triangle a){return 3*gc(a)-2*outcircle(a).o;}   //垂心
//==========================================================
double S(polygon&a)  //面积 O(n)
{
    int i; double res=0;
    a[a.n+1]=a[1];
    tr(i,1,a.n) res+=a[i]^a[i+1];
    return res/2;
}
double C(polygon&a) //周长 O(n)
{
    int i; double res=0;
    a[a.n+1]=a[1];
    tr(i,1,a.n) res+=len(a[i+1]-a[i]);
    return res;
}
int in(point a,polygon&b)  //点与多边形位置关系 0外 1内 2上 O(n)
{
    int s=0,i,d1,d2,k;
    b[b.n+1]=b[1];
    tr(i,1,b.n)
    {
        if (onseg(a,line(b[i],b[i+1]))) return 2;
        k=cmp((b[i+1]-b[i])^(b[i]-a));
        d1=cmp(b[i].y-a.y);
        d2=cmp(b[i+1].y-a.y);
        s=s+(k>0&&d2<=0&&d1>0)-(k<0&&d1<=0&&d2>0);
    }
    return s!=0;
}
point gc(polygon&a)  //重心 O(n)
{
    double s=S(a); point t(0,0); int i;
    if (cmp(s)==0) return nonp;
    a[a.n+1]=a[1];
    tr(i,1,a.n) t=t+(a[i]+a[i+1])*(a[i]^a[i+1]);
    return t/s/6;
}
int pick_on(polygon&a)  //皮克求边上格点数 O(n)
{
    int s=0,i;
    a[a.n+1]=a[1];
    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)));
    return s;
}
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)

bool __cmpx(point a,point b){return cmp(a.x-b.x)<0;}
bool __cmpy(point a,point b){return cmp(a.y-b.y)<0;}
double __mindist(point *a,int l,int r)
{
    double ans=nonx;
    if (l==r) return ans;
    int i,j,tl,tr,mid=(l+r)/2;
    ans=min(__mindist(a,l,mid),__mindist(a,mid+1,r));
    for(tl=mid;tl>=l&&a[tl].x>=a[mid].x-ans;tl--); tl++;
    for(tr=mid+1;tr<=r&&a[tr].x<=a[mid].x+ans;tr++); tr--;
    sort(a+tl+1,a+tr+1,__cmpy);
    tr(i,tl,tr-1) tr(j,i+1,min(tr,i+4)) ans=min(ans,len(a[i]-a[j]));
    sort(a+tl+1,a+tr+1,__cmpx);
    return ans;
}
double mindist(polygon&a) //a只是点集 O(nlogn)
{
    sort(a.a+1,a.a+a.n+1,__cmpx);
    return __mindist(a.a,1,a.n);
}
//line convex_maxdist(polygon a){}
bool __cmpconvex(point a,point b){return cmp(a.x-b.x)<0||(cmp(a.x-b.x)==0&&cmp(a.y-b.y)<0);}
void convex_hull(polygon&a,polygon&b) //a只是点集 O(nlogn)
{
    int i,t;
    sort(a.a+1,a.a+a.n+1,__cmpconvex);
    b.n=0;
    tr(i,1,a.n)
    {
        while (b.n>=2&&cmp((b[b.n]-b[b.n-1])^(a[i]-b[b.n-1]))<=0) b.n--;
        b[++b.n]=a[i];
    }
    t=b.n;
```

```
    rtr(i,a.n-1,1)
    {
        while (b.n>t&&cmp((b[b.n]-b[b.n-1])^(a[i]-b[b.n-1]))<=0) b.n--;
        b[++b.n]=a[i];
    }
    b.n--;
}
//int convex_in(point a,polygon b){} //0外 1内 2上 O(logn)
//polygon cross(polygon a,polygon b){}
//polygon cross(line a,polygon b){}
//double unionS(circle a,polygon b){}
circle mincovercircle(polygon&a) //最小圆覆盖 O(n)
{
    circle t; int i,j,k;
    srand(time(0));
    random_shuffle(a.a+1,a.a+a.n+1);
    for(i=2,t=circle(a[1],0);i<=a.n;i++) if (!in(a[i],t))
    for(j=1,t=circle(a[i],0);j<i;j++) if (!in(a[j],t))
    for(k=1,t=circle((a[i]+a[j])/2,len(a[i]-a[j])/2);k<j;k++) if (!in(a[k],t))
    t=outcircle(triangle(a[i],a[j],a[k]));
    return t;
}
```

4 DP

5 串

5.1 最长回文子串-Manacher

$O(N)$

```
//st，s都从1开始！
//      1 2 3 4 5 6 7 8
// st:  a b a
//  s:  0 0 a 0 b 0 a 0
//  a:  0 0 1 2 3 2 1 0
int a[2*maxl];
char st[maxl],s[2*maxl];
int manacher()
{
    int l=strlen(st+1),i,Mm,Mr=0,ans=0;
    memset(a,0,sizeof(a)); s[1]=0xFF;
    tr(i,2,2*l+2) s[i]=(i&1)*st[i/2];
    tr(i,1,2*l+2)
    {
        if (i<Mr) a[i]=min(a[2*Mm-i],Mr-i);
        while (s[i-a[i]-1]==s[i+a[i]+1]) a[i]++;
        if (i+a[i]>Mr) {Mr=i+a[i]; Mm=i;}
        ans=max(ans,a[i]);
    }
    return ans;
}
int main()
{
    gets(st+1); printf("%d\n",manacher());
    return 0;
}
```

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，$O(\sum P_i+T)$

```
//maxt=文本串长，maxp=模式串长，maxn=模式串数
struct ac{int s,to[26],fail;} a[maxn*maxp];
int m,n;
char ts[maxp],s[maxt];
queue<int> b;
void clear(int x)
{
    a[x].s=a[x].fail=0;
    memset(a[x].to,0,sizeof(a[x].to));
}
void ins(char *st)
{
    int i,x=0,c,l=strlen(st);
    tr(i,0,l-1)
    {
        if (!a[x].to[c=st[i]-'a']) {a[x].to[c]=++m; clear(m);}
        x=a[x].to[c];
    }
    a[x].s++;
}
void build()
{
    int i,h,t;
    tr(i,0,25) if (t=a[0].to[i]) b.push(t);
    while (b.size())
    {
        h=b.front(); b.pop();
        tr(i,0,25)
        if (t=a[h].to[i])
        {
            a[t].fail=a[a[h].fail].to[i];
            b.push(t);
        } else a[h].to[i]=a[a[h].fail].to[i];
    }
}
int cnt(char *st)
{
    int i,x=0,c,t,cnt=0,l=strlen(st);
    tr(i,0,l-1)
    {
        c=st[i]-'a';
        while (!a[x].to[c]&&x) x=a[x].fail;
        x=a[x].to[c];
        for(t=x;t&&a[t].s>-1;t=a[t].fail) {cnt+=a[t].s; a[t].s=-1;}
    }
    return cnt;
}
```

```
void work()
{
    int i;
    m=0; clear(0);
    scanf(”%d”,&n);
    tr(i,1,n)
    {
        scanf(”%s”,ts); ins(ts);
    }
    build();
    scanf(”%s”,s); printf(”%d\n”,cnt(s));
}
```

5.3 后缀自动机

路径对应子串
非克隆节点至少对应一个前缀
子串 endpos 为出现位置末尾下标
结点 endpos 一致
结点内子串为连续的“子串区间”(k 后缀所存在则相同
suffixlink 指向 endpos 增大最少的母集
反向 suffixlink 构树是反前缀树
endpos 集合为反向 suffixlink 能走到的所有非克隆点的 firstendpos

```
/*
注意很多字符集，特别是”-’a’”
=BASIC
    fa——————————suffix link
    to——————————trans
    mxl—————————max length
    clone———————cloned from/0
=OPTIONAL
    ac——————————accept state 是否为后缀态
    mnl—————————min length
    fep—————————first endpos 首次出现的末端
    tp——————————可供拓扑排序DP/树上DP的顺序
    sep—————————size of endpos 出现次数
    invfa———————inverse of fa(suffix-link)
=PROBLEMS
    KthSub——————第k小子串,分不同位置是否算多个两种模式(BZOJ3998)
    AllOccur————P[]在T中所有出现的位置
    LCS—————————最长公共子串
     *待补* LCS2————————多串最长公共子串
==其它
    判断P是否为T子串:T建直接跑P
    子串种数:sum{mxl[i]-mnl[i]+1}
    不同子串总长:sum{mnl[i]+..+mxl[i]}
    环串最小表示:S+S后,走最小后继一直走n次，必能走到
    P在T中首次出现位置:fep
    P在T中出现次数（可重叠）:sep
    最短非子串:DP,d[u]=1+min{d[v]},v含0
    最长重复不重叠子串:DP找出lastendpos,贪心地用fep,lep,mnl,mxl更新答案
==未想
    BZOJ3238 BZOJ2806 BZOJ2555 BZOJ3926 BZOJ3879
    BZOJ2780 BZOJ3756 BZOJ1396 HDU4622 SPOJ8222
    BZOJ4566
*/

struct sam
{
    int fa,to[chs],mxl,clone;
    int& operator[](int x){return to[x];}
    sam(int _fa=0,int _mxl=0,int _clone=0)
    {
        fa=_fa; mxl=_mxl; clone=_clone;
        memset(to,0,sizeof(to));
    }
} a[maxn<<1];
bool ac[maxn<<1];
int m,last,mnl[maxn<<1],fep[maxn<<1],sep[maxn<<1],tp[maxn<<1];
vector<int> invfa[maxn<<1];

void extend(int ch)
{
    int p=last,q,r;
    a[last=++m]=sam(p,a[p].mxl+1,0);
    for(;p&&!a[p][ch];p=a[p].fa) a[p][ch]=m;
    if (!p) {a[m].fa=1; return;}
    if (a[q=a[p][ch]].mxl==a[p].mxl+1) {a[m].fa=q; return;}
    a[r=++m]=a[q];
    a[r].clone=q;
    a[r].mxl=a[p].mxl+1;
    a[q].fa=a[last].fa=r;
    for(;p&&a[p][ch]==q;p=a[p].fa) a[p][ch]=r;
}
void getac()
{
    memset(ac,0,sizeof(ac));
    for(int x=last;x;x=a[x].fa) ac[x]=1;
}
void getmnl()
{
    int i;
    tr(i,2,m) mnl[i]=a[a[i].fa].mxl+1;
}
void getfep()
{
    int i;
    tr(i,2,m) fep[i]=a[i].clone?fep[a[i].clone]:a[i].mxl;
}
void gettp()
{
    static int ls[maxn];
    int i,mx=0;
    memset(ls,0,sizeof(ls));
    tr(i,1,m) {mx=max(mx,a[i].mxl); ls[a[i].mxl]++;}
    rtr(i,mx-1,0) ls[i]+=ls[i+1];
    tr(i,1,m) tp[ls[a[i].mxl]--]=i;
```

```
}
void getsep()
{
    gettp();
    int i;
    tr(i,1,m) sep[i]=a[i].clone?0:1;
    tr(i,1,m-1) sep[a[tp[i]].fa]+=sep[tp[i]];
}
void getinvfa()
{
    int i;
    tr(i,2,m) invfa[i].clear();
    tr(i,2,m) invfa[a[i].fa].push_back(i);
}
//========================================================
void KthSub(char *s,int k,bool diffpos)  //diffpos为1算多个
{
    static int ll d[maxn<<1];
    int n=strlen(s+1),i,j,x;
    a[m=last=1]=sam();
    tr(i,1,n) extend(s[i]-'a');
    if (diffpos) getsep();
    else gettp();
    tr(i,1,m)
    {
        x=tp[i];
        if (x!=1)
        {
            if (diffpos) d[x]=sep[x]; else d[x]=1;
        }
        tr(j,0,25) if (a[x][j]) d[x]+=d[a[x][j]];
    }
    x=1;
    if (d[1]<k) {puts("-1"); return;}
    for(;;)
    {
        ll t;
        if (x!=1)
        {
            if (diffpos) t=sep[x]; else t=1;
        } else t=0;
        if (t>=k) break;
        tr(j,0,25) if (a[x][j])
        {
            t+=d[a[x][j]];
            if (t>=k) break;
        }
        putchar('a'+j);
        t-=d[x=a[x][j]];
        k-=t;
    }
    puts("");
}
void OutputAllOccur(int x,int np)
{
    if (!a[x].clone) printf("%d ",fep[x]-np+1);
    int i,l=int(invfa[x].size())-1;
    tr(i,0,l) OutputAllOccur(invfa[x][i],np);
}
void AllOccur(char *T,int q)    //待测
{
    static char P[maxn];
    int n=strlen(T+1),np,i,j,x;
    a[m=last=1]=sam();
    tr(i,1,n) extend(T[i]-'a');
    getfep();
    getinvfa();
    tr(i,1,q)
    {
        gets(P+1);
        np=strlen(P+1);
        for(x=1,j=1;j<=np&&a[x][P[j]-'a'];j++) x=a[x][P[j]-'a'];
        if (j>np) OutputAllOccur(x,np);
        puts("");
    }
}
void LCS(char *s1,char *s2)
{
    int n1=strlen(s1+1),n2=strlen(s2+1),i,x,l,bestl,besti;
    a[m=last=1]=sam();
    tr(i,1,n1) extend(s1[i]-'a');
    x=1; l=bestl=besti=0;
    tr(i,1,n2)
    {
        while(x&&a[x][s2[i]-'a']==0) l=a[x=a[x].fa].mxl;
        if (x)
        {
            x=a[x][s2[i]-'a']; l++;
        } else x=1;
        if (l>bestl)
        {
            bestl=l; besti=i;
        }
    }
    tr(i,besti-bestl+1,besti) putchar(s2[i]);
    puts("");
}

int main()
{
    //gets(s+1); n=strlen(s+1);
    //a[m=last=1]=sam();
    //tr(i,1,n) extend(s[i]-'a');

    return 0;
}
```

6 图/树

6.1 单源最短路-Dijkstra

不加堆，$O(V^2+E)$

```
struct edge{int pre,x,y,d;} a[maxm];
int n,m,ah[maxn],d[maxn];
bool p[maxn];
void update(int x)
{
  int e;
  p[x]=true;
  for(e=ah[x];e>-1;e=a[e].pre)
    if (!p[a[e].y]&&(!d[a[e].y]||a[e].d+d[x]<d[a[e].y]))
      d[a[e].y]=a[e].d+d[x];
}
void dijkstra()
{
  int i,j,t;
    memset(p,0,sizeof(p));
  update(1);
  d[0]=oo;
  tr(i,2,n)
  {
    t=0;
    tr(j,1,n) if (!p[j]&&d[j]&&d[j]<d[t]) t=j;
    update(t);
  }
  printf("%d\n",d[n]);
}
```

加堆，$O(ElogE+V)$

```
typedef pair<int,int> pa;
struct edge{int pre,x,y,d;} a[maxm];
int n,m,ah[maxn],ans[maxn];
priority_queue<pa,vector<pa>,greater<pa> >d;
bool p[maxn];
void dijkstra()
{
  int v,s,e;
    memset(p,0,sizeof(p));
  d.push(make_pair(0,1));
  while(!d.empty())
  {
    v=d.top().second;
    s=d.top().first;
    d.pop();
    if (p[v]) continue;
    p[v]=1;
    ans[v]=s;
    for(e=ah[v];e>-1;e=a[e].pre)
      if (!p[a[e].y]) d.push(make_pair(s+a[e].d,a[e].y));
  }
  printf("%d\n",ans[n]);
}
```

6.2 最短路-Floyd

$O(V^3+E)$

```
void floyd()
{
  int i,j,k;
  tr(k,1,n) tr(i,1,n)
    if (a[i][k]) tr(j,1,n)
      if (i!=j&&a[k][j]&&(!a[i][j]||(a[i][j]&&a[i][k]+a[k][j]<a[i][j])))
            a[i][j]=a[i][k]+a[k][j];
}
```

6.3 单源最短路-SPFA

不加优化，$O(VE+V^2)=O(kE)$

```
struct edge{int pre,x,y,d;} a[maxm];
int n,m,last[maxn],d[maxn],b[maxn];
bool p[maxn];
void spfa()
{
  int h,t,e;
  memset(d,0x7F,sizeof(d));
    memset(p,0,sizeof(p));
  b[0]=1; p[1]=1; d[1]=0;
  h=n-1; t=0;
  while (h!=t)
  {
    h=(h+1)%n;
    for (e=last[b[h]];e>-1;e=a[e].pre)
      if (d[a[e].x]+a[e].d<d[a[e].y])
      {
        d[a[e].y]=d[a[e].x]+a[e].d;
        if (!p[a[e].y])
        {
          t=(t+1)%n;
          b[t]=a[e].y;
          p[a[e].y]=1;
        }
      }
    p[b[h]]=0;
  }
  printf("%d\n",d[n]);
}
```

SLF+LLL 优化，$O(VE+V^2)=O(kE)$

```
//a从1开始！
struct edge{int pre,x,y,d;} a[maxm];
int n,m,last[maxn],d[maxn],b[maxn];
bool p[maxn];
```

```
void spfa()
{
  int e,h,t,sum,num;
  memset(d,0x7F,sizeof(d));
    memset(p,0,sizeof(p));
  b[0]=1; p[1]=1; d[1]=0;
  sum=0; num=1;
  h=0; t=0;
  while (num)
  {
    while (d[h]*num>sum)    //?
    {
      t=(t+1)%n;
      b[t]=b[h];
      h=(h+1)%n;
    }
    e=last[b[h]];
    p[b[h]]=0;
    num--;
    sum-=d[a[e].x];
    h=(h+1)%n;
    for (;a[e].x;e=a[e].pre)
      if (d[a[e].x]+a[e].d<d[a[e].y])
      {
        if (p[a[e].y]) sum-=d[a[e].y];
        d[a[e].y]=d[a[e].x]+a[e].d;
        sum+=d[a[e].y];
        if (!p[a[e].y])
        {
          if (num && d[a[e].y]<d[b[h]])
          {
            h=(h+n-1)%n;
            b[h]=a[e].y;
          } else
          {
            t=(t+1)%n;
            b[t]=a[e].y;
          }
          p[a[e].y]=1;
          num++;
        }
      }
  }
  printf("%d\n",d[n]);
}
```

6.4 二分图最大匹配-匈牙利

$O(VE)$

```
struct edge{int x,y,pre;} a[maxm];
int nx,ny,m,last[maxn],my[maxn];
bool p[maxn];
int dfs(int x)
{
  for (int e=last[x];e>-1;e=a[e].pre)
    if (!p[a[e].y])
    {
      int y=a[e].y;
      p[y]=1;
      if (!my[y]||dfs(my[y])) return my[y]=x;
    }
  return 0;
}
void hungary()
{
  int i,ans=0;
  memset(my,0,sizeof(my));
  tr(i,1,nx)
  {
    memset(p,0,sizeof(p));
    if (dfs(i)) ans++;
  }
  printf("%d\n",ans);
}
```

6.5 有向图极大强连通分量-Tarjan 强连通

$O(V+E)$

```
//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
struct edge{int x,y,pre;} a[maxm];
int n,m,ah[maxn],ds,dfn[maxn],low[maxn],ss,sta[maxn],gs,group[maxn],size[maxn
    ];
bool insta[maxn];
void tarjan(int x)
{
    int e,y,t;
    dfn[x]=low[x]=++ds;
    sta[++ss]=x; insta[x]=1;
    for(e=ah[x];e>-1;e=a[e].pre)
    {
        if (!dfn[y=a[e].y])
        {
            tarjan(y);
            low[x]=min(low[x],low[y]);
        }
        else if (insta[y]) low[x]=min(low[x],dfn[y]);
    }
    if (low[x]==dfn[x])
        for(gs++,t=0;t!=x;t=sta[ss--])
        {
            group[sta[ss]]=gs;
            size[gs]++;
            insta[sta[ss]]=0;
        }
}
void work()
{
```

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

6.6 最大流-iSAP

简版（无 BFS，递归，gap，cur），$O(V^2 * E)$

```
struct edge{int x,y,c,f,pre;} a[2*maxm];
int n,mm,m,last[maxn],d[maxn],gap[maxn],cur[maxn],ans;
void newedge(int x,int y,int c,int f)
{
  m++;
  a[m].x=x; a[m].y=y; a[m].c=c; a[m].f=f;
  a[m].pre=last[x]; last[x]=m;
}
void init()
{
  int i,x,y,c;
  m=-1;
  memset(last,-1,sizeof(last));
  tr(i,1,mm)
  {
    x=read(); y=read(); c=read();
    newedge(x,y,c,0);
    newedge(y,x,c,c);
  }
  tr(i,1,n) cur[i]=last[i];
    memset(d,0,sizeof(d));
  memset(gap,0,sizeof(gap));
  gap[0]=n;
  ans=0;
}
int sap(int x,int flow)
{
  int e,t;
  if (x==n) return flow;
  for (e=cur[x];e!=-1;e=a[e].pre)
    if (a[e].f<a[e].c && d[a[e].y]+1==d[x])
    {
      cur[x]=e;
      if (t=sap(a[e].y,min(flow,a[e].c-a[e].f)))
      {
        a[e].f+=t; a[e^1].f-=t; return t;
      }
    }
  if (--gap[d[x]]==0) d[n]=n;
  d[x]=n;
  for (e=last[x];e!=-1;e=a[e].pre)
    if (a[e].f<a[e].c) d[x]=min(d[x],d[a[e].y]+1);
  cur[x]=last[x];
  ++gap[d[x]];
  return 0;
}
int work()
{
    while (d[n]<n) ans+=sap(1,oo);
}
```

完全版（有 BFS，非递归，gap，cur），$O(V^2 * E)$

```
int n,mm,m,ans,last[maxn],cur[maxn],pre[maxn],d[maxn],gap[maxn],b[maxn];
bool p[maxn];
struct edge{int x,y,c,f,pre;} a[2*maxm];
void newedge(int x,int y,int c,int f)
{
  m++;
  a[m].x=x; a[m].y=y; a[m].c=c; a[m].f=f;
  a[m].pre=last[x]; last[x]=m;
}
void init()
{
  int i,x,y,c;
  m=-1;
  memset(last,-1,sizeof(last));
  tr(i,1,mm)
  {
    x=read(); y=read(); c=read();
    newedge(x,y,c,0);
    newedge(y,x,c,c);
  }
}
int aug()
{
  int x,flow=a[cur[1]].c-a[cur[1]].f;
  for (x=pre[n];x>1;x=pre[x]) flow=min(flow,a[cur[x]].c-a[cur[x]].f);
  return flow;
}
void bfs()
{
  int h,t,e;
  memset(p,0,sizeof(p));
  b[1]=n; p[n]=1;
  h=0; t=1;
  while (h<t)
  {
    h++;
    for (e=last[b[h]];e!=-1;e=a[e].pre)
      if (a[e].c==a[e].f && !p[a[e].y])
      {
        b[++t]=a[e].y;
        p[a[e].y]=1;
        d[a[e].y]=d[a[e].x]+1;
      }
  }
}
void sap()
{
  int x,e,flow;
  memset(d,0,sizeof(d));
  memset(gap,0,sizeof(gap));
```

```
  bfs();
  tr(x,1,n) gap[d[x]]++;
  ans=0;
  tr(x,1,n) cur[x]=last[x];
  x=1; pre[1]=1;
  while (d[1]<n)
  {
    for (e=cur[x];e!=-1;e=a[e].pre)
      if (d[x]==d[a[e].y]+1 && a[e].f<a[e].c)
      {
        cur[x]=e;
        pre[a[e].y]=x;
        x=a[e].y;
        break;
      }
    if (e==-1)
    {
      if (!(--gap[d[x]])) return;
      cur[x]=last[x];
      d[x]=n;
      for (e=last[x];e!=-1;e=a[e].pre)
        if (a[e].f<a[e].c) d[x]=min(d[x],d[a[e].y]+1);
      gap[d[x]]++;
      x=pre[x];
    }
    if (x==n){
      flow=aug();
      for (x=pre[x];x>1;x=pre[x])
      {
        a[cur[x]].f+=flow; a[cur[x]^1].f-=flow;
      }
      a[cur[x]].f+=flow; a[cur[x]^1].f-=flow;
      ans+=flow;
      x=1;
    }
  }
}
```

6.7 最小生成树-Prim

不加堆，$O(V+E)$

```
struct edge{int x,y,d,pre;} a[maxm];
int n,m,ah[maxn],d[maxn];
bool p[maxn];
void prim()
{
  int i,j,x,y,e,ans=0;
  memset(d,0x7f,sizeof(d)); d[1]=0;
  memset(p,0,sizeof(p));
  tr(i,1,n)
  {
    x=0;
    tr(j,1,n) if (!p[j]&&d[j]<d[x]) x=j;
    ans+=d[x];
    p[x]=1;
    for(e=ah[x];e>-1;e=a[e].pre)
      if (!p[y=a[e].y]) d[y]=min(d[y],a[e].d);
  }
  printf("%d\n",ans);
}
```

加堆，$O(V+E)$

```
struct edge{int x,y,d,pre;} a[maxm];
typedef pair<int,int> pa;
priority_queue<pa,vector<pa>,greater<pa> >d;
int n,m,ah[maxn];
bool p[maxn];
void prim()
{
  int i,x,y,e,ans=0;
  pa t;
  while (!d.empty()) d.pop();
  d.push(make_pair(0,1));
  memset(p,0,sizeof(p));
  tr(i,1,n)
  {
        while (!d.empty()&&p[d.top().second]) d.pop();
    t=d.top();
    ans+=t.first;
    p[x=t.second]=1;
    for(e=ah[x];e>-1;e=a[e].pre)
      if (!p[y=a[e].y]) d.push(make_pair(a[e].d,y));
  }
  printf("%d\n",ans);
}
```

6.8 最小生成树-Kruskal

$O(ElogE+E)$

```
//a从1开始！
struct edge{int x,y,d;} a[maxm];
bool cmp(edge a,edge b){return a.d<b.d;}
int n,i,j,m,fa[maxn];
int gfa(int x){return x==fa[x]?x:fa[x]=gfa(fa[x]);}
void kruskal()
{
  int ans,fx,fy;
  sort(a+1,a+m+1,cmp);
  tr(i,1,n) fa[i]=i;
  ans=0;
  tr(i,1,m)
    if ((fx=gfa(a[i].x))!=(fy=gfa(a[i].y)))
    {
      fa[fx]=fy;
      ans+=a[i].d;
    }
  printf("%d\n",ans);
```

```
}
```

6.9 树的直径-BFS

$O(N)$

```
struct edge{int x,y,d,pre;} a[2*maxn];
int n,m,ah[maxn],d0[maxn],d1[maxn],b[maxn];
bool p[maxn];
void bfs(int root,int *d)
{
  int h,t,e,y;
  memset(p,0,sizeof(p));
  h=0; t=1;
  b[1]=root;
  p[root]=1;
  while (h<t)
  {
    h++;
    for (e=ah[b[h]];e>-1;e=a[e].pre)
      if (!p[y=a[e].y])
      {
        b[++t]=y;
        p[y]=1;
        d[y]=d[a[e].x]+a[x].d;
      }
  }
}
void work()
{
  int i,s1,s2;
    memset(d0,0,sizeof(d0));
  memset(d1,0,sizeof(d1));
  bfs(1,d0); s1=1; tr(i,1,n) if (d0[i]>d0[s1]) s1=i;
  bfs(s1,d1); s2=1; tr(i,1,n) if (d1[i]>d1[s2]) s2=i;
    printf("%d %d %d\n",s1,s2,d1[s2]);
}
```

6.10 LCA-TarjanLCA

$O(N+Q)$

```
struct query{int x,y,pre,lca;} b[2*maxq];
struct edge{int x,y,pre,d;} a[2*maxn];
int n,q,am,bm,ah[maxn],bh[maxn],fa[maxn],dep[maxn];
bool p[maxn];
int gfa(int x){return fa[x]==x?x:fa[x]=gfa(fa[x]);}
void tarjan(int x,int depth)
{
    int tmp,y;
    p[x]=1;
    dep[x]=depth;
    for(tmp=ah[x];tmp>-1;tmp=a[tmp].pre)
        if (!p[y=a[tmp].y])
        {
            tarjan(y,depth+a[tmp].d);
            fa[y]=x;
        }
    for(tmp=bh[x];tmp>-1;tmp=b[tmp].pre)
        if (p[y=b[tmp].y]) b[tmp].lca=b[tmp^1].lca=gfa(y);
}
void work()
{
    memset(dep,0,sizeof(dep));
    memset(p,0,sizeof(p));
    tarjan(1,0);
    int i; tr(i,0,q-1) writeln(dep[b[2*i].x]+dep[b[2*i].y]-2*dep[b[2*i].lca]);
}
```

7 数据结构

7.1 并查集

```
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

$O(NlogN+QlogN)$

```
int n,a[maxn],f[maxn];
char tc;
void add(int x,int y)
{
    while (x<=n) {f[x]+=y; x+=x&-x;}
}
int sum(int x)
{
    int res=0;
    while (x) {res+=f[x]; x-=x&-x;}
    return res;
}
void work()
{
    int q,i,tx,ty;
    n=read(); q=read();
    memset(f,0,sizeof(f));
    tr(i,1,n) add(i,a[i]=read());
    tr(i,1,q)
    {
        tc=getchar(); tx=read(); ty=read();
        if (tc=='M') {add(tx,ty-a[tx]); a[tx]=ty;}
        else writeln(sum(ty)-sum(tx-1));
    }
}
```

7.3 区间和 _ 区间修改单点查询-树状数组

$O(NlogN+QlogN)$

```
int n,i,f[maxn];
void add(int x,int y)
{
    while (x) {f[x]+=y; x-=x&-x;}
}
int sum(int x)
{
    int res=0;
    while (x<=n) {res+=f[x]; x+=x&-x;}
    return res;
}
void work()
{
    int q,i;
    n=read(); q=read();
    memset(f,0,sizeof(f));
    tr(i,1,q)
    {
        tc=getchar();
        if (tc=='M') {add(read()-1,-1); add(read(),1);}
        else writeln(sum(read()));
    }
}
```

7.4 区间和 _ 区间修改区间查询-树状数组

差分得数列 $d_i=a_i-a_{i-1}$，则

$$\sum_{i=1}^{x}a_i=(x+1)\sum_{i=1}^{x}d_i-\sum_{i=1}^{x}i\cdot d_i$$

$O(NlogN+QlogN)$

```
int n,m,i,b[maxn],x,y,z;
ll a[maxn],A[maxn];
char ch;

void add(ll *f,int x,ll y)
{
    while (x<=n) {f[x]+=y; x+=x&-x;}
}
ll sum(ll *f,int x)
{
    ll res=0;
    while (x) {res+=f[x]; x-=x&-x;}
    return res;
}
ll ask(int x)
{
  return 1LL*(x+1)*sum(a,x)-sum(A,x);
}
int main()
{
  scanf("%d%d",&n,&m);
  tr(i,1,n) scanf("%d",&b[i]);
  rtr(i,n,1) b[i]=b[i]-b[i-1];
  tr(i,1,n)
  {
    add(a,i,b[i]); add(A,i,1LL*b[i]*i);
  }
  tr(i,1,m)
  {
    for(ch=getchar();ch!='C'&&ch!='Q';ch=getchar());
    if (ch=='Q')
    {
      scanf("%d%d",&x,&y);
      printf("%lld\n",ask(y)-ask(x-1));
    } else
    {
      scanf("%d%d%d",&x,&y,&z);
      add(a,x,z); add(A,x,1LL*z*x);
      if (y<n)
      {
        add(a,y+1,-z); add(A,y+1,-1LL*z*(y+1));
      }
    }
  }
  return 0;
}
```

7.5 区间和-线段树

$O(NlogN+QlogN)$

```
struct node{int s,tag;} a[4*maxn];
int n;
void update(int t,int l,int r)
{
    if (l!=r)
    {
        a[t<<1].tag+=a[t].tag;
        a[t<<1|1].tag+=a[t].tag;
    }
    a[t].s+=(int)(r-l+1)*a[t].tag;
    a[t].tag=0;
}
void add(int t,int l,int r,int x,int y,int z)
{
    if (x<=l&&r<=y) {a[t].tag+=z; return ;}
    a[t].s+=(int)(min(r,y)-max(l,x)+1)*z;
    update(t,l,r);
    int mid=(l+r)>>1;
    if (x<=mid) add(t<<1,l,mid,x,y,z);
    if (y>mid) add(t<<1|1,mid+1,r,x,y,z);
}
```

```
int sum(int t,int l,int r,int x,int y)
{
    int res=0;
    update(t,l,r);
    if (x<=l&&r<=y) return a[t].s;
    int mid=(l+r)>>1;
    if (x<=mid) res+=sum(t<<1,l,mid,x,y);
    if (y>mid) res+=sum(t<<1|1,mid+1,r,x,y);
    return res;
}
void work()
{
    int q,i,tx,ty; char tc;
    n=read(); q=read();
    tr(i,1,n) add(1,1,n,i,i,read());
    tr(i,1,q)
    {
        tc=getchar(); tx=read(); ty=read();
        if (tc=='A') add(1,1,n,tx,ty,read());
        else writeln(sum(1,1,n,tx,ty));
    }
}
```

7.6 平衡树-Treap

$O(QlogN)$

```
struct node
{
    node* ch[2];
    int x,y,size;
    int chsize(int d){return ch[d]?ch[d]->size:0;}
} ;
node *root;
void newnode(node *&t,int x)
{
    t=new node;
    t->ch[0]=t->ch[1]=0;
    t->x=x; t->y=rand(); t->size=1;
}
void rot(node *&t,int d)
{
    node *tt=t->ch[!d];
    t->ch[!d]=tt->ch[d];
    tt->ch[d]=t;
    tt->size=t->size;
    t->size=t->chsize(0)+t->chsize(1)+1;
    t=tt;
}
void ins(node *&t,int x)
{
    if (!t) newnode(t,x);
    else {
        int d=t->x<x; ins(t->ch[d],x); ++t->size;
        if (t->ch[d]->y<t->y) rot(t,!d);
    }
}
void del(node *&t,int x)
{
    if (x==t->x)
    {
        if (!t->ch[0]||!t->ch[1])
        {
            node *tt=t; t=t->ch[t->ch[0]==0]; delete(tt);
            return;
        } else
        {
            int d=t->ch[0]->y<t->ch[1]->y;
            rot(t,d); del(t->ch[d],x);
        }
    } else del(t->ch[t->x<x],x);
    --t->size;
}
node* kth(node *&t,int k)
{
    if (k<=t->chsize(0)) return kth(t->ch[0],k);
    else if(k>t->chsize(0)+1) return kth(t->ch[1],k-(t->chsize(0)+1));
    else return t;
}
void work()
{
    srand(time(0)); newnode(root,oo);
    //...
}
```

7.7 平衡树-Splay

$O(QlogN)$

```
struct node
{
    int x,ch[2],fa,size,cnt;
    int& operator[](int t){return ch[t];}
} a[maxm];
int m,root,err;

void newnode(int x)
{
    a[++m].x=x;
    a[m][0]=a[m][1]=a[m].fa=0;
    a[m].size=a[m].cnt=1;
}
void addnode(int t,int x=1){a[t].size+=x;a[t].cnt+=x;}
void dl(int t){if(!t)return;}    //download
void ul(int t){if(!t)return;a[t].size=a[a[t][0]].size+a[a[t][1]].size+a[t].cnt;}    //upload
void dlk(int p,int q,int&rt){dl(p);if(q)a[p][a[q].x>a[p].x]=a[q].fa=0;}  //del link p->q
void blk(int p,int q,int&rt){if(q)a[(p?a[p][a[q].x>a[p].x]:rt)=q].fa=p;ul(p);}  //build link p->q
```

```
//注意从上至下dlk，从下至上blk
void rot(int t,int&rt)     //单旋
{
    int p=a[t].fa,q=a[p].fa,r=a[t][a[p][0]==t];
    dlk(q,p,rt); dlk(p,t,rt); dlk(t,r,rt);
    blk(p,r,rt); blk(t,p,rt); blk(q,t,rt);
}
void splay(int t,int&rt)
{
    for(int p;t!=rt;rot(t,rt))
        if ((p=a[t].fa)!=rt) rot(t==a[p][0]^p==a[a[p].fa][0]?t:p,rt);      //
            双旋
}
void find(int x,int&rt,int type=0)    //-1:pre(<=)    0:normal(pre/suc)    1:
    suc(>=)
{
    int s=0,t=rt;
    for(;t;t=(a[t].x!=x)?a[t][x>a[t].x]:0)
        if (type*(a[t].x-x)>=0) s=t;
    if (!s) {err=1; return;}
    splay(s,rt);
}
void kth(int k,int&rt)
{
    if (k<1||k>a[rt].size) {err=1; return;}
    int t=rt;
    for(k-=a[a[rt][0]].size;k<1||k>a[t].cnt;k-=(k>0?a[t].cnt:-a[a[t][0]].size)
        +a[a[t=a[t][k>0]][0]].size);
    splay(t,rt);
}
void ins(int x,int&rt)
{
    find(x,rt);
    if (a[rt].x==x) return addnode(rt);
    int p=a[rt][x>a[rt].x];
    dlk(rt,p,rt);
    newnode(x); blk(m,p,rt); blk(rt,m,rt);
}
void del(int x,int&rt,int all=0)
{
    find(x,rt);
    if (a[rt].x!=x) {err=1; return;}
    if (!all&&a[rt].cnt>1) return addnode(rt,-1);
    if (!a[rt][0]) return blk(0,a[rt][1],rt);
    find(a[rt].x,a[rt][0]);
    int p=a[rt][0],q=a[rt][1];
    dlk(rt,p,rt); dlk(rt,q,rt);
    blk(p,q,rt); blk(0,p,rt);
}
void work()
{
    newnode(oo); root=1;
    //...
}
```

7.8 区间第 k 大 _ 无修改-主席树

$O(NlogN + QlogN)$

```
struct node{int l,r,size;} a[maxm];
int n,q,m,num,b[maxn],dc[maxn],root[maxn];
int rdc(int x){return lower_bound(dc+1,dc+num+1,x)-dc;}
void init()
{
    int i;
    n=read(); q=read();
    tr(i,1,n) b[i]=read();
    memcpy(dc,b,(n+1)*sizeof(int));
    sort(dc+1,dc+n+1);
    num=unique(dc+1,dc+n+1)-(dc+1);
}
int insert(int tx,int l,int r,int x)
{
    int t,mid=(l+r)>>1;
    a[t=++m]=a[tx]; a[t].size++;
    if (l==r) return t;
    if (x<=mid) a[t].l=insert(a[tx].l,l,mid,x);
    else a[t].r=insert(a[tx].r,mid+1,r,x);
    return t;
}
int kth(int tx,int ty,int l,int r,int k)
{
    int ds,mid=(l+r)>>1;
    if (l==r) return l;
    if (k<=(ds=a[a[ty].l].size-a[a[tx].l].size))
        return kth(a[tx].l,a[ty].l,l,mid,k);
    else return kth(a[tx].r,a[ty].r,mid+1,r,k-ds);
}
void work()
{
    int i,x,y,z;
    tr(i,1,n) root[i]=insert(root[i-1],1,num,rdc(b[i]));
    tr(i,1,q)
    {
        x=read(); y=read(); z=read();
        writeln(dc[kth(root[x-1],root[y],1,num,z)]);
    }
}
```

7.9 区间第 k 大 _ 有修改-树状数组套主席树

$O(NlogN + QlogNlogN)$

```
//C是初始数列根（数组），c是修改根（树状数组）
int n,m,q,num,b[maxn],c[maxn],dc[maxn+maxq],sx,sy,lx[maxn],ly[maxn];
int C[maxn];
struct node{int l,r,size;} a[maxm];
struct oper{int type,x,y,z;} op[maxq];

int lowbit(int x){return x&(-x);}
```

```
int rdc(int x){return lower_bound(dc+1,dc+num+1,x)-dc;}
int update(int t,int l,int r,int x,int y)
{
  int tt=t,mid=(l+r)>>1;
  a[tt=++m]=a[t];
  a[tt].size+=y;
  if (l==r) return tt;
  if (x<=mid) a[tt].l=update(a[t].l,l,mid,x,y);
  else a[tt].r=update(a[t].r,mid+1,r,x,y);
  return tt;
}
void init()
{
  int i,x,y,z;
  scanf(”%d%d”,&n,&q);
  tr(i,1,n) {scanf(”%d”,&b[i]); dc[i]=b[i];}
  tr(i,1,q)
  {
    char ch;
    for(ch=getchar();ch!=’C’&&ch!=’Q’;ch=getchar());
    if (ch==’C’)
    {
      scanf(”%d%d”,&x,&y);
      op[i].type=1;
      op[i].x=x; op[i].y=y;
      dc[n+i]=y;
    } else
    {
      scanf(”%d%d%d”,&x,&y,&z);
      op[i].type=2;
      op[i].x=x; op[i].y=y; op[i].z=z;
      dc[n+i]=0;
    }
  }
  sort(dc+1,dc+n+q+1);
    num=unique(dc+1,dc+n+q+1)-(dc+1);
    tr(i,1,q) if (op[i].type==1) op[i].y=rdc(op[i].y);
    tr(i,1,n) b[i]=rdc(b[i]);
    //
    m=1;
  C[0]=1;
    tr(i,1,n) C[i]=update(C[i-1],1,num,b[i],1);
    //
    tr(i,1,n) c[i]=1;
}
int kth(int l,int r,int k)
{
  if (l==r) return l;
  int i,sz=0,mid=(l+r)>>1;
  tr(i,1,sx) sz-=a[a[lx[i]].l].size;
  tr(i,1,sy) sz+=a[a[ly[i]].l].size;
  if (k<=sz)
  {
    tr(i,1,sx) lx[i]=a[lx[i]].l;
    tr(i,1,sy) ly[i]=a[ly[i]].l;
    return kth(l,mid,k);
  } else
  {
    tr(i,1,sx) lx[i]=a[lx[i]].r;
    tr(i,1,sy) ly[i]=a[ly[i]].r;
    return kth(mid+1,r,k-sz);
  }
}
void work()
{
  int i,x,y,z,t;
  tr(i,1,q)
    if (op[i].type==1)
    {
      x=op[i].x; y=op[i].y;
      for(t=x;t<=n;t+=lowbit(t))
        c[t]=update(c[t],1,num,b[x],-1);
      b[x]=y;
      for(t=x;t<=n;t+=lowbit(t))
        c[t]=update(c[t],1,num,b[x],1);
    } else
    {
      x=op[i].x-1; y=op[i].y; z=op[i].z;
      for(sx=0,t=x;t;t-=lowbit(t)) lx[++sx]=c[t];
      lx[++sx]=C[x];
      for(sy=0,t=y;t;t-=lowbit(t)) ly[++sy]=c[t];
      ly[++sy]=C[y];
      printf(”%d\n”,dc[kth(1,num,z)]);
    }
}
```

7.10 RMQ-ST

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

```
//！！注意！！__builtin_clz只有g++能用
//x为int时，31-__builtin_clz(x) 等价于 int(log(x)/log(2))
//x为ll时，63-__builtin_clzll(x) 等价于 (ll)(log(x)/log(2))
int n,q,mn[maxn][maxln];
void init()
{
    int i;
    n=read(); q=read();
    tr(i,1,n) mn[i][0]=read();
}
void st()
{
    int i,j,ln;
    ln=31-__builtin_clz(n);
    tr(i,1,ln) tr(j,1,n-(1<<i)+1)
        mn[j][i]=min(mn[j][i-1],mn[j+(1<<(i-1))][i-1]);
}
void work()
{
    int i,x,y,t;
    st();
```

```
    tr(i,1,q)
    {
        x=read(); y=read();
        t=31-__builtin_clz(y-x+1);
        writeln(min(mn[x][t],mn[y-(1<<t)+1][t]));
    }
}
```

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

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

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

■ 和差化积

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

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

■ 积化和差

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

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

■ 二、三、n 倍角（切比雪夫）

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

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

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

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

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

■ **二、三次降幂**

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

■ **万能公式**

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

■ **连乘**

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

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

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

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

■ **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**

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

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

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

■ **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

$\genfrac{[}{]}{0pt}{}{n}{1} = (n-1)! \qquad \genfrac{[}{]}{0pt}{}{n}{2} = (n-1)!H_{n-1} \qquad \genfrac{[}{]}{0pt}{}{n}{n-1} = \binom{n}{2} \qquad \sum_{k=0}^{n} \genfrac{[}{]}{0pt}{}{n}{k} = n!$

$\genfrac{[}{]}{0pt}{}{n+1}{m+1} = \sum_k \genfrac{[}{]}{0pt}{}{n}{k} \binom{k}{m} = n! \sum_{k=0}^{n} \frac{1}{k!} \genfrac{[}{]}{0pt}{}{k}{m}$

$\genfrac{[}{]}{0pt}{}{n}{m} = \sum_k \genfrac{[}{]}{0pt}{}{n+1}{k+1} \binom{k}{m} (-1)^{m-k} \qquad \genfrac{[}{]}{0pt}{}{m+n+1}{m} = \sum_{k=0}^{m} k(n+k) \genfrac{[}{]}{0pt}{}{n+k}{k}$

$\genfrac{[}{]}{0pt}{}{n}{\ell+m} \binom{\ell+m}{\ell} = \sum_k \genfrac{[}{]}{0pt}{}{k}{\ell} \genfrac{[}{]}{0pt}{}{n-k}{m} \binom{n}{k}$

■ 第二类斯特林数

$\genfrac{\{}{\}}{0pt}{}{n}{k}$ 表第二类斯特林数，表基数为 n 的集合的 k 份划分方法数，

$$\genfrac{\{}{\}}{0pt}{}{n}{1} = \genfrac{\{}{\}}{0pt}{}{n}{n} = 1, \genfrac{\{}{\}}{0pt}{}{n}{k} = \genfrac{\{}{\}}{0pt}{}{n-1}{k-1} + k \genfrac{\{}{\}}{0pt}{}{n-1}{k}$$

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

$\genfrac{\{}{\}}{0pt}{}{n}{2} = 2^{n-1} - 1 \qquad \genfrac{\{}{\}}{0pt}{}{n}{n-1} = \binom{n}{2} \qquad \genfrac{[}{]}{0pt}{}{n}{k} \geq \genfrac{\{}{\}}{0pt}{}{n}{k}$

$\genfrac{\{}{\}}{0pt}{}{n+1}{m+1} = \sum_k \binom{n}{k} \genfrac{\{}{\}}{0pt}{}{k}{m} = \sum_{k=0}^{n} \genfrac{\{}{\}}{0pt}{}{k}{m} (m+1)^{n-k}$

$\genfrac{\{}{\}}{0pt}{}{n}{m} = \sum_k \binom{n}{k} \genfrac{\{}{\}}{0pt}{}{k+1}{m+1} (-1)^{n-k} \qquad \genfrac{\{}{\}}{0pt}{}{m+n+1}{m} = \sum_{k=0}^{m} k \genfrac{\{}{\}}{0pt}{}{n+k}{k}$

$\binom{n}{m} = \sum_k \genfrac{\{}{\}}{0pt}{}{n+1}{k+1} \genfrac{[}{]}{0pt}{}{k}{m} (-1)^{m-k}$

$(n-m)!\binom{n}{m} = \sum_k \genfrac{[}{]}{0pt}{}{n+1}{k+1} \genfrac{\{}{\}}{0pt}{}{k}{m} (-1)^{m-k}, \quad \forall n \geq m$

$\genfrac{\{}{\}}{0pt}{}{n}{n-m} = \sum_k \binom{m-n}{m+k} \binom{m+n}{n+k} \genfrac{[}{]}{0pt}{}{m+k}{k}$

$\genfrac{[}{]}{0pt}{}{n}{n-m} = \sum_k \binom{m-n}{m+k} \binom{m+n}{n+k} \genfrac{\{}{\}}{0pt}{}{m+k}{k}$

$\genfrac{\{}{\}}{0pt}{}{n}{\ell+m} \binom{\ell+m}{\ell} = \sum_k \genfrac{\{}{\}}{0pt}{}{k}{\ell} \genfrac{\{}{\}}{0pt}{}{n-k}{m} \binom{n}{k}$

■ 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

$B_n = \sum_{k=1}^{n} \genfrac{\{}{\}}{0pt}{}{n}{k} \qquad B_n = \frac{1}{e} \sum_{k=0}^{\infty} \frac{k^n}{k!} \qquad \sum_{n=0}^{\infty} \frac{B_n}{n!} x^n = e^{e^x - 1}$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

9.1.4 导数

■ **几个导数**

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

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

■ **高阶导数**

(莱布尼茨公式)

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

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

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

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

9.1.5 积分表

■ $ax + b (a \neq 0)$

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

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

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

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

■ $\sqrt{ax+b}$

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

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

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

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

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

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

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. $\int\frac{\mathrm{d}x}{x^2\sqrt{x^2-a^2}}=\frac{\sqrt{x^2-a^2}}{a^2x}+C$
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. $\int\frac{\sqrt{x^2-a^2}}{x}\mathrm{d}x=\sqrt{x^2-a^2}-a\arccos\frac{a}{|x|}+C$
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. $\int\frac{\mathrm{d}x}{x^2\sqrt{a^2-x^2}}=-\frac{\sqrt{a^2-x^2}}{a^2x}+C$
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. $\int \ln x\mathrm{d}x = x\ln x - x + C$
2. $\int \frac{\mathrm{d}x}{x\ln x} = \ln\big|\ln x\big| + C$
3. $\int x^n\ln x\mathrm{d}x = \frac{1}{n+1}x^{n+1}(\ln x - \frac{1}{n+1}) + C$
4. $\int (\ln x)^n\mathrm{d}x = x(\ln x)^n - n\int(\ln x)^{n-1}\mathrm{d}x$
5. $\int x^m(\ln x)^n\mathrm{d}x = \frac{1}{m+1}x^{m+1}(\ln x)^n - \frac{n}{m+1}\int x^m(\ln x)^{n-1}\mathrm{d}x$

■ **三角函数**

1. $\int \sin x\mathrm{d}x = -\cos x + C$
2. $\int \cos x\mathrm{d}x = \sin x + C$
3. $\int \tan x\mathrm{d}x = -\ln|\cos x| + C$
4. $\int \cot x\mathrm{d}x = \ln|\sin x| + C$
5. $\int \sec x\mathrm{d}x = \ln\left|\tan\left(\frac{\pi}{4}+\frac{x}{2}\right)\right| + C = \ln|\sec x + \tan x| + C$
6. $\int \csc x\mathrm{d}x = \ln\left|\tan\frac{x}{2}\right| + C = \ln|\csc x - \cot x| + C$
7. $\int \sec^2 x\mathrm{d}x = \tan x + C$
8. $\int \csc^2 x\mathrm{d}x = -\cot x + C$
9. $\int \sec x\tan x\mathrm{d}x = \sec x + C$
10. $\int \csc x\cot x\mathrm{d}x = -\csc x + C$
11. $\int \sin^2 x\mathrm{d}x = \frac{x}{2} - \frac{1}{4}\sin 2x + C$
12. $\int \cos^2 x\mathrm{d}x = \frac{x}{2} + \frac{1}{4}\sin 2x + C$
13. $\int \sin^n x\mathrm{d}x = -\frac{1}{n}\sin^{n-1}x\cos x + \frac{n-1}{n}\int\sin^{n-2}x\mathrm{d}x$
14. $\int \cos^n x\mathrm{d}x = \frac{1}{n}\cos^{n-1}x\sin x + \frac{n-1}{n}\int\cos^{n-2}x\mathrm{d}x$
15. $\int \frac{\mathrm{d}x}{\sin^n x} = -\frac{1}{n-1}\frac{\cos x}{\sin^{n-1}x} + \frac{n-2}{n-1}\int\frac{\mathrm{d}x}{\sin^{n-2}x}$
16. $\int \frac{\mathrm{d}x}{\cos^n x} = \frac{1}{n-1}\frac{\sin x}{\cos^{n-1}x} + \frac{n-2}{n-1}\int\frac{\mathrm{d}x}{\cos^{n-2}x}$
17. $$\begin{aligned}&\int\cos^m x\sin^n x\mathrm{d}x\\ =&\frac{1}{m+n}\cos^{m-1}x\sin^{n+1}x + \frac{m-1}{m+n}\int\cos^{m-2}x\sin^n x\mathrm{d}x\\ =&-\frac{1}{m+n}\cos^{m+1}x\sin^{n-1}x + \frac{n-1}{m+1}\int\cos^m x\sin^{n-2}x\mathrm{d}x\end{aligned}$$
18. $\int \sin ax\cos bx\mathrm{d}x = -\frac{1}{2(a+b)}\cos(a+b)x - \frac{1}{2(a-b)}\cos(a-b)x + C$
19. $\int \sin ax\sin bx\mathrm{d}x = -\frac{1}{2(a+b)}\sin(a+b)x + \frac{1}{2(a-b)}\sin(a-b)x + C$
20. $\int \cos ax\cos bx\mathrm{d}x = \frac{1}{2(a+b)}\sin(a+b)x + \frac{1}{2(a-b)}\sin(a-b)x + C$
21. $\int\frac{\mathrm{d}x}{a+b\sin x} = \begin{cases}\frac{2}{\sqrt{a^2-b^2}}\arctan\frac{a\tan\frac{x}{2}+b}{\sqrt{a^2-b^2}} + C & (a^2>b^2)\\ \frac{1}{\sqrt{b^2-a^2}}\ln\left|\frac{a\tan\frac{x}{2}+b-\sqrt{b^2-a^2}}{a\tan\frac{x}{2}+b+\sqrt{b^2-a^2}}\right| + C & (a^2<b^2)\end{cases}$
22. $\int\frac{\mathrm{d}x}{a+b\cos x} = \begin{cases}\frac{2}{a+b}\sqrt{\frac{a+b}{a-b}}\arctan\left(\sqrt{\frac{a-b}{a+b}}\tan\frac{x}{2}\right) + C & (a^2>b^2)\\ \frac{1}{a+b}\sqrt{\frac{a+b}{a-b}}\ln\left|\frac{\tan\frac{x}{2}+\sqrt{\frac{a+b}{b-a}}}{\tan\frac{x}{2}-\sqrt{\frac{a+b}{b-a}}}\right| + C & (a^2<b^2)\end{cases}$
23. $\int\frac{\mathrm{d}x}{a^2\cos^2 x + b^2\sin^2 x} = \frac{1}{ab}\arctan\left(\frac{b}{a}\tan x\right) + C$
24. $\int\frac{\mathrm{d}x}{a^2\cos^2 x - b^2\sin^2 x} = \frac{1}{2ab}\ln\left|\frac{b\tan x+a}{b\tan x-a}\right| + C$
25. $\int x\sin ax\mathrm{d}x = \frac{1}{a^2}\sin ax - \frac{1}{a}x\cos ax + C$
26. $\int x^2\sin ax\mathrm{d}x = -\frac{1}{a}x^2\cos ax + \frac{2}{a^2}x\sin ax + \frac{2}{a^3}\cos ax + C$
27. $\int x\cos ax\mathrm{d}x = \frac{1}{a^2}\cos ax + \frac{1}{a}x\sin ax + C$
28. $\int x^2\cos ax\mathrm{d}x = \frac{1}{a}x^2\sin ax + \frac{2}{a^2}x\cos ax - \frac{2}{a^3}\sin ax + C$

■ **反三角函数** $(a>0)$

1. $\int \arcsin\frac{x}{a}\mathrm{d}x = x\arcsin\frac{x}{a}+\sqrt{a^2-x^2}+C$
2. $\int x\arcsin\frac{x}{a}\mathrm{d}x = (\frac{x^2}{2}-\frac{a^2}{4})\arcsin\frac{x}{a}+\frac{x}{4}\sqrt{x^2-x^2}+C$
3. $\int x^2\arcsin\frac{x}{a}\mathrm{d}x = \frac{x^3}{3}\arcsin\frac{x}{a}+\frac{1}{9}(x^2+2a^2)\sqrt{a^2-x^2}+C$
4. $\int \arccos\frac{x}{a}\mathrm{d}x = x\arccos\frac{x}{a}-\sqrt{a^2-x^2}+C$
5. $\int x\arccos\frac{x}{a}\mathrm{d}x = (\frac{x^2}{2}-\frac{a^2}{4})\arccos\frac{x}{a}-\frac{x}{4}\sqrt{a^2-x^2}+C$
6. $\int x^2\arccos\frac{x}{a}\mathrm{d}x = \frac{x^3}{3}\arccos\frac{x}{a}-\frac{1}{9}(x^2+2a^2)\sqrt{a^2-x^2}+C$
7. $\int \arctan\frac{x}{a}\mathrm{d}x = x\arctan\frac{x}{a}-\frac{a}{2}\ln(a^2+x^2)+C$
8. $\int x\arctan\frac{x}{a}\mathrm{d}x = \frac{1}{2}(a^2+x^2)\arctan\frac{x}{a}-\frac{a}{2}x+C$
9. $\int x^2\arctan\frac{x}{a}\mathrm{d}x = \frac{x^3}{3}\arctan\frac{x}{a}-\frac{a}{6}x^2+\frac{a^3}{6}\ln(a^2+x^2)+C$

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 $\sum\limits_{i=1}^{n}2^{-d_i}\le 1$，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**

$S=\frac{1}{2}ab\sin C=\frac{a^2\sin B\sin C}{2\sin(B+C)}=\sqrt{p(p-a)(p-b)(p-c)}=\frac{1}{2}\left\|\begin{matrix}a_x & a_y & 1\\ b_x & b_y & 1\\ c_x & c_y & 1\end{matrix}\right\|$，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** $\mathrm{d}s=\sqrt{[x'(t)]^2+[y'(t)]^2}\mathrm{d}t=\sqrt{1+[f'(x)]^2}\mathrm{d}x=\sqrt{r^2(\theta)+[r'(\theta)]^2}\mathrm{d}\theta$

■ **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** $V=\frac{1}{3}h(S_1+\sqrt{S_1S_2}+S_2)$

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

■ **四面体体积**
$V=\frac{1}{6}\left\|\begin{matrix}p_x & p_y & p_z\\ q_x & q_y & q_z\\ r_x & r_y & r_z\end{matrix}\right\|$，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 $V=\pi\int_a^b f^2(x)\mathrm{d}x$
侧面积 $F=2\pi\int f(x)\mathrm{d}s=2\pi\int_a^b\sqrt{1+[f'(x)]^2}\mathrm{d}x$
（空心）质心

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

（空心）转动惯量

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

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 $(0<x\le m)$，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 );
size_t strspn ( const char * str1, const char * str2 );
size_t strcspn ( const char * str1, const char * str2 );
const char * strpbrk ( const char * str1, const char * str2 );
      char * strpbrk (       char * str1, const char * str2 );
char * strtok ( char * str, const char * delimiters );
// 1. strrchr的搜索包括\0，所以strrchr(s,0)返回末尾指针
// 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
// 3. strpbrk返回str1中最先出现在str2中的字符的指针
// 4. strtok通过delimitters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
//——————————————————————其它——————————————————————
size_t strlen ( const char * str );
```

10.1.4 <cstdlib>

```
//——————————————————————转换——————————————————————
double atof (const char* str);
int atoi (const char * str);
char *  itoa ( int value, char * str, int base );
// 1. atof，atoi和itoa的十进制支持符号，itoa支持科学计数
// 2. itoa非标！Linux下没有！
//——————————————————————内存——————————————————————
void* malloc (size_t size);
void free (void* ptr);
// 1. e.g.    int *p=(int*)malloc(100*sizeof(int));
//            ...
//            free(p);
// 2. 退出函数不自动释放
//——————————————————————其它——————————————————————
int abs (int n);
void srand (unsigned int seed);
int rand (void);
void qsort (void* base, size_t num, size_t size,
            int (*compar)(const void*,const void*));
void* bsearch (const void* key, const void* base,
               size_t num, size_t size,
               int (*compar)(const void*,const void*));
// 1. rand产生最大的数是RAND_MAX，Linux下是2^31-1，Win下是32767
// 2. qsort中compar返回负，则前一个变量排在前，0和正类此
// 3. bsearch为二分，只能找base有没有key，基本废的
```

10.1.5 无头文件

```
//——————————————————————函数数组——————————————————————
//e.g. int (*cmp[])(char*a,char*b)={cmp0,cmp1,cmp2,cmp3};
```

10.2 C++

10.2.1 <iostream>/<ios>

```
//——————————————————————输入——————————————————————
istream& get (char& c);
istream& get (char* s, streamsize n);
istream& get (char* s, streamsize n, char delim);
istream& getline (char* s, streamsize n );
istream& getline (char* s, streamsize n, char delim );
istream& read (char* s, streamsize n);
istream& ignore (streamsize n = 1, int delim = EOF);
int peek();
istream& putback (char c);
streampos tellg();
istream& seekg (streampos pos);
bool eof() const;
// 1. get保留'/0'，而getline不保留
// 2. ignore一直忽略字符，直到够n个或遇到delim停止
// 3. peek偷窥下一个字符
// 4. streampos要用ll存，e.g.: t=cin.tellg(); ...; cin.seekg(t);
// 5. while (!cin.eof(...))
//——————————————————————输出——————————————————————
fmtflags setf (fmtflags fmtfl);
void unsetf (fmtflags mask);
streamsize precision (streamsize prec);
// cout.precision(...); cout<<...;
// 此处为精度，若设置fixed，则为小数点后保留位数
// 对于以下，e.g. cout<<dec<<...;
// 以下三个取消效果为函数名前加no
ios_base& uppercase (ios_base& str);
ios_base& showpos (ios_base& str);
ios_base& showpoint (ios_base& str);
// 以下两个的取消效果，应用cout.unsetf(ios_base::floatfield);
ios_base& scientific (ios_base& str);
ios_base& fixed (ios_base& str);
ios_base& dec (ios_base& str);
```

```
ios_base& hex (ios_base& str);
ios_base& oct (ios_base& str);
```

10.2.2 Containers

```
//注意很多时候iterator表区间首末时为左闭右开区间
//————————————————vector————————————————
iterator begin();
iterator end();
size_type size() const;
reference operator[] (size_type n);
reference front();
reference back();
void push_back (const value_type& val);
void pop_back();
iterator erase (iterator position);
iterator erase (iterator first, iterator last);
void clear();
//sort(v.begin(),v.end());可以实现排序
//————————————————queue————————————————
size_type size() const;
value_type& front();
void push (const value_type& val);
void pop();
//————————————————deque————————————————
size_type size() const;
reference front();
reference back();
void push_front (const value_type& val);
void push_back (const value_type& val);
void pop_front();
void pop_back();
void clear();
priority_queue
template <class T, class Container = vector<T>, class Compare = less<typename Container::value_type> > class priority_queue;
//默认Compare=less，即为大根堆
size_type size() const;
const value_type& top() const;
void push (const value_type& val);
void pop();
//————————————————set————————————————
// /multiset/map/multimap 类似
template < class T, class Compare = less<T>, class Alloc = allocator<T> > class set;
iterator begin();
iterator end();
//end是最后元素的下一个，即空
size_type size() const;
iterator find (const value_type& val) const;
//找不到返回end()
size_type count (const value_type& val) const;
//0/1，除非是 multiset/multimap
iterator lower_bound (const value_type& val) const;
//大于等于val的最小一个
iterator upper_bound (const value_type& val) const;
pair insert (const value_type& val);
void erase (iterator position);
size_type erase (const value_type& val);
void erase (iterator first, iterator last);
void clear();
//在 map/multimap里，at()只定义在 C++11，但operator[]任用
//————————————————unordered_set————————————————
// 仅C++11! iterator被认为无太大意义而舍去 template < class Key, class Hash = hash<Key>, class Pred = equal_to<Key>, class Alloc = allocator<Key> > class unordered_set;
size_type size() const noexcept;
iterator find ( const key_type& k );
size_type count ( const key_type& k ) const;
//0/1
pair insert ( const value_type& val );
size_type erase ( const key_type& k );
void clear() noexcept;
size_type bucket_count() const noexcept;
float load_factor() const noexcept;
float max_load_factor() const noexcept;
void max_load_factor ( float z );
void rehash ( size_type n );
```

10.2.3 <string>

```
//  string的+, =, +=, ==, !=, <, <=, >, >=都对cstring, char, string支持
//————————————————迭代器————————————————
      iterator begin();
const_iterator begin() const;
      iterator end();
const_iterator end() const;
      reverse_iterator rbegin();
const_reverse_iterator rbegin() const;
      reverse_iterator rend() noexcept;
const_reverse_iterator rend() const noexcept;
//  1.  get保留'/0'，而getline不保留
//————————————————长度————————————————
size_t size() const;
bool empty() const;
void resize (size_t n);
void resize (size_t n, char c);
void clear();
//  1.  resize时若n大于原长，c指定则用c填充尾部至n，否则填充'\0'
//————————————————修改————————————————
string& append (const string& str);
string& append (const string& str, size_t subpos, size_t sublen);
string& append (const char* s);
string& append (const char* s, size_t n);
string& append (size_t n, char c);
template <class InputIterator>
    string& append (InputIterator first, InputIterator last);

```

```
string& assign (const string& str);
string& assign (const string& str, size_t subpos, size_t sublen);
string& assign (const char* s);
string& assign (const char* s, size_t n);
string& assign (size_t n, char c);
template <class InputIterator>
   string& assign (InputIterator first, InputIterator last);

string& insert (size_t pos, const string& str);
string& insert (size_t pos, const string& str, size_t subpos, size_t sublen);
string& insert (size_t pos, const char* s);
string& insert (size_t pos, const char* s, size_t n);
string& insert (size_t pos, size_t n, char c);
void insert (iterator p, size_t n, char c);
iterator insert (iterator p, char c);
template <class InputIterator>
   void insert (iterator p, InputIterator first, InputIterator last);

string& erase (size_t pos = 0, size_t len = npos);
iterator erase (iterator p);
iterator erase (iterator first, iterator last);

string& replace (size_t pos,  size_t len,  const string& str);
string& replace (iterator i1, iterator i2, const string& str);
string& replace (size_t pos,  size_t len,  const string& str,
                 size_t subpos, size_t sublen);
string& replace (size_t pos,  size_t len,  const char* s);
string& replace (iterator i1, iterator i2, const char* s);
string& replace (size_t pos,  size_t len,  const char* s, size_t n);
string& replace (iterator i1, iterator i2, const char* s, size_t n);
string& replace (size_t pos,  size_t len,  size_t n, char c);
string& replace (iterator i1, iterator i2, size_t n, char c);
template <class InputIterator>
  string& replace (iterator i1, iterator i2,
                   InputIterator first, InputIterator last);

void swap (string& str);
//———————————————————————查询———————————————————————
const char* c_str() const;
const char* data() const;
//c_str()后有'\0'而data()没有，但data()效率更高
//以下找不到返回string::npos
size_t find (const string& str, size_t pos = 0) const;
size_t find (const char* s, size_t pos = 0) const;
size_t find (const char* s, size_t pos, size_t n) const;
size_t find (char c, size_t pos = 0) const;
//rfind只找pos开始或pos之前开始的
size_t rfind (const string& str, size_t pos = npos) const;
size_t rfind (const char* s, size_t pos = npos) const;
size_t rfind (const char* s, size_t pos, size_t n) const;
size_t rfind (char c, size_t pos = npos) const;
size_t find_first_of (const string& str, size_t pos = 0) const;
size_t find_first_of (const char* s, size_t pos = 0) const;
size_t find_first_of (const char* s, size_t pos, size_t n) const;
size_t find_first_of (char c, size_t pos = 0) const;
size_t find_last_of (const string& str, size_t pos = npos) const;
size_t find_last_of (const char* s, size_t pos = npos) const;
size_t find_last_of (const char* s, size_t pos, size_t n) const;
size_t find_last_of (char c, size_t pos = npos) const;
size_t copy (char* s, size_t len, size_t pos = 0) const;
string substr (size_t pos = 0, size_t len = npos) const;
```

10.2.4 pb_ds

```
using namespace __gnu_pbds
//———————————————————————堆———————————————————————
#include<ext/pb_ds/priority_queue.hpp>
template<
    typename Value_Type,
    typename Cmp_Fn = std::less<Value_Type>,
    typename Tag = pairing_heap_tag,
    typename Allocator = std::allocator<char> >
class priority_queue;
size_type size () const
bool empty () const
void clear ()
point_iterator push (const_reference r_val)
void pop ()
const_reference top () const
void modify (point_iterator it, const_reference r_new_val)
void erase (point_iterator it)
template<class Pred> size_type erase_if (Pred prd)
void join (priority_queue &other)
template<class Pred> void split (Pred prd, priority_queue &other)
iterator begin ()
const_iterator begin () const
iterator end ()
const_iterator end () const
//  1. Tag：pairing_heap_tag, binary_heap_tag, binomial_heap_tag,
    rc_binomial_heap_tag, thin_heap_tag五种。常用前两种，binary_heap只能代替
    queue里的priority_queue（只push，pop，top等，否则线性），更复杂则采用
    pairing_heap。
//  2. 注意一定要__gnu_pbds::priority_queue，不然会歧义报错
//  3. erase_if返回删除个数
//  4. 关于Pred的用例：
//         bool p(int x){return x&1;}
//         a.erase_if(p);
//  5. join的other会被清空
//———————————————————————树———————————————————————
#include<ext/pb_ds/assoc_container.hpp>
#include<ext/pb_ds/tree_policy.hpp>
template<
    typename Key,
    typename Mapped,
    typename Cmp_Fn = std::less<Key>,
    typename Tag = rb_tree_tag,
    template<
        typename Const_Node_Iterator,
        typename Node_Iterator,
        typename Cmp_Fn_,
```

```
        typename Allocator_>
    class Node_Update = null_tree_node_update,
    typename Allocator = std::allocator<char> >
class tree;
iterator lower_bound (const_key_reference r_key)
const_iterator lower_bound (const_key_reference r_key) const
iterator upper_bound (const_key_reference r_key)
const_iterator upper_bound (const_key_reference r_key) const
iterator erase (iterator it)
reverse_iterator erase (reverse_iterator it)
reverse_iterator rbegin ()
const_reverse_iterator rbegin () const
reverse_iterator rend ()
const_reverse_iterator rend () const
void join (basic_tree &other)
void split (const_key_reference r_key, basic_tree &other)
node_iterator node_begin ()
const_node_iterator node_begin () const
node_iterator node_end ()
const_node_iterator node_end () const
//  1.  Mapped设置为null_type（较旧版本为null_mapped_type）变set，即对iterator加星取值
//  2.  Tag：rb_tree_tag, splay_tree_tag, or ov_tree_tag三种，一般只用rb_tree
//  3.  Node_Update自带有tree_order_statistics_node_update，包含find_by_order和order_of_key两个函数，order从0开始，order_of_key找比x严格小的个数
//————————————————————哈希————————————————————
template<
    typename Key,
    typename Mapped,
    typename Hash_Fn = std::hash<Key>,
    typename Eq_Fn = std::equal_to<Key>,
    typename Comb_Hash_Fn = direct_mask_range_hashing<>
    typename Resize_Policy = default explained below.
     bool Store_Hash = false,
     typename Allocator = std::allocator<char> >
class cc_hash_table;
template<
    typename Key,
    typename Mapped,
    typename Hash_Fn = std::hash<Key>,
    typename Eq_Fn = std::equal_to<Key>,
    typename Comb_Probe_Fn = direct_mask_range_hashing<>
    typename Probe_Fn = default explained below.
    typename Resize_Policy = default explained below.
    bool Store_Hash = false,
    typename Allocator =  std::allocator<char> >
class gp_hash_table;
```

10.2.5 无头文件

```
//————————————————————函数重载————————————————————
//  以sort举例
struct t{int a;...} ar[n];
//重载operator<
bool operator<(const t&x,const t&y){return x.a<y.a;}
sort(a,a+n);
//比较函数
bool cmp(const t&x,const t&y){return x.a<y.a;}
sort(a,a+n,cmp);
//仿函数1（重载operator()）
struct cmp(){bool operator()(const t&x,const t&y){return x.a<y.a;}}
sort(a,a+n,cmp());
//仿函数2（重载operator()）
struct cmp(){bool operator()(const t&x,const t&y){return x.a<y.a;}} p;
sort(a,a+n,p);
```

11 经典错误

= 和 == 混淆；scanf 没加 &；爆数组/数据范围