Contents

1 Basic

1.1 default code

```
#include <bits/stdc++.h>
#define PB push_back
#define MP make_pair
#define F first
#define S second
#define SZ(x) ((int)(x).size())
#define ALL(x) (x).begin(),(x).end()
#ifdef _DEBUG_
  #define debug(...) printf(__VA_ARGS__)
#else
  #define debug(...) (void)0
#endif
using namespace std;
typedef long long ll;
typedef pair<int,int> PII;
typedef vector<int> VI;

int main() {
  return 0;
}
```

1.2 .vimrc

```
color torte
syn on
set guifont=Consolas:h16: nu sc ai si ts=4 sm sts=4 sw=4

map <F9> <ESC>:w<CR>:!g++ % -o %< -O2 -Wall -Wno-unused-result -std=c++0x<CR>
map <S-F9> <ESC>:w<CR>:!g++ % -o %< -O2 -Wall -Wno-unused-result -D_DEBUG_ -std=c++0x<CR>
map <F5> <ESC>:!./%<<CR>
map <F6> <ESC>:w<CR>ggvG"+y
map <S-F5> <ESC>:!./%< < %<.in<CR>
imap <Home> <ESC>^i
com INPUT sp %<.in
```

2 math

2.1 ext gcd

```
// find one solution (x,y) of ax+by=gcd(a,b)
void ext_gcd(int a,int b,int &g,int &x,int &y)
{
  if(!b){ g=a; x=1; y=0; }
  else{ ext_gcd(b, a%b, g, y, x); y -= x*(a/b); }
}
```

2.2 FFT other

```cpp
/* FFT code from shik in CodeForces*/
/* zj a577*/
#include <bits/stdc++.h>
using namespace std;
const int N=300000;

const double PI=acos(-1.0);
struct vir{
    double re,im;
    vir( double _re=0, double _im=0 ):re(_re),im(_im){}
};
vir operator +( vir a, vir b ) { return vir(a.re+b.re,a.im+b .im); }
vir operator -( vir a, vir b ) { return vir(a.re-b.re,a.im-b .im); }
vir operator *( vir a, vir b ) { return vir(a.re*b.re-a.im*b .im,a.re*b.im+a.im*b.re); }
vir x1[2*N],x2[2*N];

int rev( int x, int len ) {
    int r=0,i;
    for ( i=0; i<len; i++,x>>=1 ) r=(r<<1)+(x&1);
    return r;
}
void change( vir *x, int len, int loglen ) {
    for ( int i=0; i<len; i++ )
        if ( rev(i,loglen)<i ) swap(x[rev(i,loglen)],x[i]);
}
void fft( vir *x, int len, int loglen ) {
    change(x,len,loglen);
    int i,j,s,t=1;
    for ( i=0; i<loglen; i++,t<<=1 ) {
        for ( s=0; s<len; s+=t+t ) {
            vir a,b,wo(cos(PI/t),sin(PI/t)),wn(1,0);
            for ( j=s; j<s+t; j++ ) {
                a=x[j]; b=x[j+t]*wn;
                x[j]=a+b; x[j+t]=a-b;
                wn=wn*wo;
            }
        }
    }
}
void dit_fft( vir *x, int len, int loglen ) {
    int i,j,s,t=len>>1;
    for ( i=0; i<loglen; i++,t>>=1 ) {
        for ( s=0; s<len; s+=t+t ) {
            vir a,b,wn(1,0),wo(cos(PI/t),-sin(PI/t));
            for ( j=s; j<s+t; j++ ) {
                a=x[j]+x[j+t]; b=(x[j]-x[j+t])*wn;
                x[j]=a; x[j+t]=b;
                wn=wn*wo;
            }
        }
    }
    change(x,len,loglen);
    for ( i=0; i<len; i++ ) x[i].re/=len;
}

char a[N],b[N];
int ans[2*N];

int main()
{
  int na,nb,len=1,loglen=0;
  while(scanf(”%s%s”,a,b)==2)
  {
    for(int i=2*N-1;i>=0;i--)
      x1[i]=x2[i]=0.0;
    for(na=0;a[na];na++);
    for(nb=0;b[nb];nb++);
    for(int i=na-1;i>=0;i--)
      x1[i]=(double)(a[na-i-1]-’0’);
    for(int i=nb-1;i>=0;i--)
      x2[i]=(double)(b[nb-i-1]-’0’);
    while(len<=2*max(na,nb)+5)
    {
      len*=2;
      loglen++;
    }
    fft(x1,len,loglen);
    fft(x2,len,loglen);
    for(int i=0;i<len;i++)
      x1[i]=x1[i]*x2[i];
    dit_fft(x1,len,loglen);
    for(int i=len-1;i>=0;i--)
      ans[i]=(int)round(x1[i].re+0.01);
    for(int i=0;i<len;i++)
    {
      if(ans[i]>=10)
      {
        ans[i+1]+=ans[i]/10;
        ans[i]%=10;
      }
    }
    bool zero=0;
    for(int i=len-1;i>=0;i--)
    {
      //printf(”%d\n”,ans[i]);
      if(zero)
        printf(”%d”,ans[i]);
      else if(ans[i]>0)
      {
        printf(”%d”,ans[i]);
        zero=1;
      }
    }
    if(!zero)
      printf(”0”);
    puts(””);
  }
  return 0;
}
```

2.3 MillerRabin other

```cpp
/* Miller Rabin code from ioicamp */
#include <bits/stdc++.h>
#define PB push_back
#define MP make_pair
```

```cpp
#define F first
#define S second
#define SZ(x) ((int)(x).size())
#define ALL(x) (x).begin(),(x).end()
#ifdef _DEBUG_
  #define debug(...) printf(__VA_ARGS__)
#else
  #define debug(...) 0
#endif
using namespace std;
typedef long long ll;
typedef pair<int,int> PII;
typedef vector<int> VI;

ll mul(ll a, ll b, ll n) {
  ll r = 0;
  a %= n, b %= n;
  while(b) {
    if(b&1) r = (a+r>=n ? a+r-n : a+r);
    a = (a+a>=n ? a+a-n : a+a);
    b >>= 1;
  }
  return r;
}

ll bigmod(ll a, ll d, ll n) {
  if(d==0)  return 1LL;
  if(d==1) return a % n;
  return mul(bigmod(mul(a, a, n), d/2, n),
      d%2?a:1, n);
}

const bool PRIME = 1, COMPOSITE = 0;
bool miller_rabin(ll n, ll a) {
  if(__gcd(a, n) == n)  return PRIME;
  if(__gcd(a, n) != 1)  return COMPOSITE;
  ll d = n-1, r = 0, res;
  while(d%2==0) { ++r; d/=2; }
  res = bigmod(a, d, n);
  if(res == 1 || res == n-1)  return PRIME;
  while(r--) {
    res = mul(res, res, n);
    if(res == n-1)  return PRIME;
  }
  return COMPOSITE;
}

bool isprime(ll n) {
  if(n==1)
    return COMPOSITE;
  ll as[7] = {2, 325, 9375, 28178, 450775,
      9780504, 1795265022};
  for(int i=0; i<7; i++)
    if(miller_rabin(n, as[i]) == COMPOSITE)
        return COMPOSITE;
  return PRIME;
}
```

3 flow

3.1 dinic

```cpp
#include <bits/stdc++.h>
#define PB push_back
#define MP make_pair
#define F first
#define S second
#define SZ(x) ((int)(x).size())
using namespace std;
typedef long long ll;
typedef pair<int,int> PII;
typedef vector<int> VI;

/*******************************/
// dinic
const int MAXV=300;
const int MAXE=10000;
const int INF=(int)1e9+10;

struct E{
  int to,co;//capacity
  E(int t=0,int c=0):to(t),co(c){}
}eg[2*MAXE];

// source:0  sink:n-1
struct Flow{
  VI e[MAXV];
  int ei,v;
  void init(int n) {
    v=n;
    ei=0;
    for(int i=0;i<n;i++)
      e[i]=VI();
  }
  void add(int a,int b,int c) { //a to b ,
      maxflow=c
    eg[ei]=E(b,c);
    e[a].PB(ei);
    ei++;
    eg[ei]=E(a,0);
    e[b].PB(ei);
    ei++;
  }

  int d[MAXV],qu[MAXV],ql,qr;
  bool BFS() {
    memset(d,-1,v*sizeof(int));
    ql=qr=0;
    qu[qr++]=0;
    d[0]=0;
    while(ql<qr && d[v-1]==-1) {
      int n=qu[ql++];
      VI &v=e[n];
      for(int i=v.size()-1;i>=0;i--) {
        int u=v[i];
        if(d[eg[u].to]==-1 && eg[u].co>0) {
          d[eg[u].to]=d[n]+1;
          qu[qr++]=eg[u].to;
        }
      }
    }
    return d[v-1]!=-1;
  }
  int ptr[MAXV];
  int go(int n,int p) {
    if(n==v-1)
      return p;
    VI &u=e[n];
    int temp;
    for(int i=ptr[n];i<SZ(u);i++)
    {
      if(d[n]+1!=d[eg[u[i]].to] || eg[u[i]].co==0)
        continue;
      if((temp=go(eg[u[i]].to,min(p,eg[u[i]].co)))==0)
        continue;
      eg[u[i]].co-=temp;
      eg[u[i]^1].co+=temp;
      ptr[n]=i;
      return temp;
    }
    ptr[n]=SZ(u);
    return 0;
  }
  int max_flow() {
    int ans=0,temp;
    while(BFS()) {
      for(int i=0;i<v;i++)
        ptr[i]=0;
      while((temp=go(0,INF))>0)
        ans+=temp;
    }
    return ans;
  }
}flow;

int main() {

  return 0;
}
```

4 string

4.1 KMP

```
/***
Test OJ 265
trivial string matching

input:
abc
abccbabbabc

output:
0 8

***/
#include <bits/stdc++.h>
#define PB push_back
#define F first
#define S second
#define SZ(x) ((int)(x).size())
#define MP make_pair
using namespace std;
typedef long long ll;
typedef pair<int,int> PII;
typedef vector<int> VI;

char S[500010],T[500010];
int K[500010];

int main()
{
  gets(S);
  gets(T);
  K[0]=-1;
  int a=-1;
  for(int i=1;S[i];i++)
  {
    while(a!=-1 && S[a+1]!=S[i])
      a=K[a];
    if(S[a+1]==S[i])
      a++;
    K[i]=a;
  }
  VI ans;
  a=-1;
  for(int i=0;T[i];i++)
  {
    while(a!=-1 && S[a+1]!=T[i])
      a=K[a];
    if(S[a+1]==T[i])
      a++;
    if(!S[a+1])
    {
      ans.PB(i-a);
      a=K[a];
    }
  }
  bool first=1;
  for(int u:ans)
  {
    if(first)
      printf(”%d”,u),first=0;
    else
      printf(” %d”,u);
  }
  puts(””);
  return 0;
}
```

4.2 Z-value

```
/***
Test OJ 265
trivial string matching

input:
abc
abccbabbabc

output:
0 8

***/
#include <bits/stdc++.h>
#define pb push_back
#define F first
#define S second
#define SZ(x) ((int)(x).size())
#define MP make_pair
using namespace std;
```

```
typedef long long ll;
typedef pair<int,int> PII;
typedef vector<int> VI;

char S[1000010];
int Z[1000010];

int main()
{
  int len=0,lenS;
  gets(S);
  for(;S[len];len++);
  lenS=len;
  gets(S+len+1);
  for(len++;S[len];len++);
  S[len]='*';
  int bst=0;
  Z[0]=0;
  for(int i=1;i<len;i++)
  {
    if(Z[bst]+bst<i) Z[i]=0;
    else Z[i]=min(Z[bst]+bst-i,Z[i-bst]);
    while(S[Z[i]]==S[i+Z[i]]) Z[i]++;
    if(Z[i]+i>Z[bst]+bst) bst=i;
  }
  bool first=1;
  for(int i=lenS+1;i<len;i++)
    if(Z[i]>=lenS)
    {
      if(first)
        printf("%d",i-lenS-1),first=0;
      else
        printf(" %d",i-lenS-1);
    }
  puts("");
  return 0;
}
```

4.3 Suffix Array($O(NlogN)$)

```
// NTUJ448
#include <bits/stdc++.h>
#define pb push_back
#define F first
#define S second
#define SZ(x) ((int)(x).size())
#define MP make_pair
using namespace std;
typedef long long ll;
typedef pair<int,int> PII;
typedef vector<int> VI;

const int SASIZE=2500000;
char in[500];
int S[SASIZE],from[SASIZE];
int R[SASIZE],SA[SASIZE],H[SASIZE];
int tR[SASIZE],tSA[SASIZE];
int cnt[SASIZE];
int num[4010];

int main()
{
  int N;
 while(scanf("%d",&N)==1 && N)
 {
   int len=0,maxR=0;
   for(int i=0;i<N;i++)
   {
     scanf("%s",in);
     for(int j=0;in[j];j++)
     {
       from[len]=i;
       S[len++]=in[j]-'a';
     }
     from[len]=N;
     S[len++]=i+50;
   }
   memset(R,-1,sizeof(R));
   memset(cnt,0,sizeof(cnt));
   for(int i=0;i<len;i++)
   {
     R[i]=S[i];
     maxR=max(maxR,R[i]);
   }
   for(int i=0;i<len;i++)
     cnt[R[i]+1]++;
   for(int i=1;i<=maxR;i++)
     cnt[i]+=cnt[i-1];
   for(int i=0;i<len;i++)
     SA[cnt[R[i]]++]=i;
 /*  for(int i=0;i<len;i++)
     printf("R[%d]=%d, SA[%d]=%d\n",i,R[i
        ],i,SA[i]);*/
   for(int i=1;i<len;i*=2)
   {
     memset(cnt,0,sizeof(int)*(maxR+10));
     memcpy(tSA,SA,sizeof(int)*(len+10));
     memcpy(tR,R,sizeof(int)*(len+i+10));
     for(int j=0;j<len;j++)
       cnt[R[j]+1]++;
     for(int j=1;j<=maxR;j++)
       cnt[j]+=cnt[j-1];
     for(int j=len-i;j<len;j++)
       SA[cnt[R[j]]++]=j;
     for(int j=0;j<len;j++)
     {
       int k=tSA[j]-i;
       if(k<0)
         continue;
       SA[cnt[R[k]]++]=k;
     }
     int num=0;
     maxR=0;
     R[SA[0]]=num;
     for(int j=1;j<len;j++)
     {
       if(tR[SA[j-1]]<tR[SA[j]] || tR[SA[j
          -1]+i]<tR[SA[j]+i])
         num++;
       R[SA[j]]=num;
       maxR=max(maxR,R[SA[j]]);
     }
   /*  puts("-------");
     for(int i=0;i<len;i++)
       printf("R[%d]=%d, SA[%d]=%d\n",i,R[
          i],i,SA[i]);*/
   }
   memset(H,0,sizeof(H));
```

```
    for(int i=0;i<len;i++)
    {
      if(R[i]==0)
        continue;
      int &t=H[R[i]];
      if(i>0)
        t=max(0,H[R[i-1]]-1);
      while(S[i+t]==S[SA[R[i]-1]+t]) t++;
    }
    /*for(int i=0;i<len;i++)
      printf("R[%d]=%d, SA[%d]=%d\n",i,R[i
          ],i,SA[i]);
    for(int i=0;i<len;i++)
      printf("%3d %3d  %s\n",H[i],SA[i],S+
          SA[i]);*/
    /*for(int i=0;i<len;i++)
    {
      printf("%3d %3d %d|",H[i],SA[i],from[
          i]);
      for(int j=SA[i];j<len;j++)
        printf("%2d ",S[j]);
      puts("");
    }*/
    memset(num,0,sizeof(num));
    int anslen=0,ansfrom=-1;
    int get=0;
    deque<PII> deq;
  /*  for(int i=0;i<len;i++)
      printf("%d:%d\n",i,from[i]);*/
    for(int l=0,r=0;r<len;r++)
    {
      if(from[SA[r]]<N && num[from[SA[r
          ]]]==0)
        get++;
      num[from[SA[r]]]++;
      while(deq.size()>0 && deq.back().F>=H
          [r]) deq.pop_back();
      deq.pb(MP(H[r],r));
      while(num[from[SA[l]]]>1)
      {
        num[from[SA[l]]]--;
        l++;
      }
      while(deq.size()>0 && deq.front().S<=
          l) deq.pop_front();
      if(get==N && deq.front().F>anslen)
        anslen=deq.front().F, ansfrom=SA[l
            ];
    }
    //printf("(%d)\n",anslen);
    if(anslen==0)
      puts("IDENTITY LOST");
    else
    {
      for(int i=ansfrom;i<ansfrom+anslen;i
          ++)
        putchar(S[i]+'a');
      puts("");
    }
  }
  return 0;
}
```

4.4 Aho-Corasick

```
#include <cstdio>
#include <cstring>
#include <new>

struct Trie{
  int c;
  Trie *fail,*ch[52];
  Trie():c(0){memset(ch,0,sizeof(ch));}
}trie[1000100];

char m[1010],f[100100];
Trie *str[1010],*na,*root;

inline int c_i(char a)
{
  return (a>='A' && a<='Z') ? a-'A' : a-'a'
      +26;
}

void insert(char *s,int num)
{
  Trie *at=root;
  while(*s)
  {
    if(!at->ch[c_i(*s)])
      at->ch[c_i(*s)]=new (na++) Trie();
    at=at->ch[c_i(*s)],s++;
  }
  str[num]=at;
}

Trie *q[1000100];
int ql,qr;

void init()
{
  ql=qr=-1;
  q[++qr]=root;
  root->fail=NULL;
  while(ql<qr)
  {
    Trie *n=q[++ql],*f;
    for(int i=0;i<52;i++)
    {
      if(!n->ch[i])
        continue;
      f=n->fail;
      while(f && !f->ch[i])
        f=f->fail;
      n->ch[i]->fail=f?f->ch[i]:root;
      q[++qr]=n->ch[i];
    }
  }
}

void go(char *s)
{
  Trie*p=root;
  while(*s)
  {
    while(p && !p->ch[c_i(*s)])
      p=p->fail;
```

```cpp
    p=p?p->ch[c_i(*s)]:root;
    p->fi=1;
    s++;
  }
}

void AC()
{
  for(int i=qr;i>0;i--)
    q[i]->fail->c+=q[i]->c;
}

int main()
{
  int T,q;
  scanf("%d",&T);
  while(T--)
  {
    na=trie;
    root=new (na++) Trie();
    scanf("%s",f);
    scanf("%d",&q);
    for(int i=0;i<q;i++)
    {
      scanf("%s",m);
      insert(m,i);
    }
    init();
    go(f);
    for(int i=0;i<q;i++)
      puts(str[i]->fi?"y":"n");
  }
  return 0;
}
```

5 graph

5.1 Bipartite matching($O(N^3)$)

```cpp
// NTUJ1263
#include <bits/stdc++.h>
#define pb push_back
#define F first
#define S second
#define SZ(x) ((int)(x).size())
#define MP make_pair
using namespace std;
typedef long long ll;
typedef pair<int,int> PII;
typedef vector<int> VI;

bool is(ll x)
{
  ll l=1,r=2000000,m;
  while(l<=r)
  {
    m=(l+r)/2;
    if(m*m==x)
      return 1;
    if(m*m<x)
      l=m+1;
    else
      r=m-1;
  }
  return 0;
}

VI odd,even;
int in[300];
VI e[300];
int match[300];
bool vis[300];

bool DFS(int x)
{
  vis[x]=1;
  for(int u:e[x])
  {
    if(match[u]==-1 || (!vis[match[u]]&&DFS(match[u])))
    {
      match[u]=x;
      match[x]=u;
      return 1;
    }
  }
  return 0;
}

int main()
{
  int N;
  while(scanf("%d",&N)==1)
  {
    odd.clear();
    even.clear();
    for(int i=0;i<N;i++)
      e[i].clear();
    for(int i=0;i<N;i++)
    {
      scanf("%d",in+i);
      if(in[i]%2==0)
        even.pb(i);
      else
        odd.pb(i);
    }
    for(int i:even)
      for(int j:odd)
        if(is(1ll*in[i]*in[i]+1ll*in[j]*in[j]) && __gcd(in[i],in[j])==1)
          e[i].pb(j), e[j].pb(i);
    int ans=0;
    fill(match,match+N,-1);
    for(int i=0;i<N;i++)
      if(match[i]==-1)
      {
        fill(vis,vis+N,0);
        if(DFS(i))
          ans++;
      }
    printf("%d\n",ans);
  }
  return 0;
}
```

6 data structure

6.1 Treap

```cpp
#include <cstdlib>
#include <cstdio>
#include <algorithm>

using namespace std;

typedef long long ll;

const int N = 100000 + 10;

struct Treap {
  static Treap mem[N], *pmem;

  int sz, pri;
  ll val, sum, add;
  Treap *l, *r;

  Treap() {}
  Treap(ll _val):
    l(NULL), r(NULL), sz(1), pri(rand()), val(_val), sum(_val), add(0) {}
} Treap::mem[N], *Treap::pmem = Treap::mem;

Treap* make(ll val) {
  return new (Treap::pmem++) Treap(val);
}

inline int sz(Treap *t) {
  return t ? t->sz : 0;
}

inline ll sum(Treap *t) {
  return t ? t->sum + t->add * sz(t) : 0;
}

inline void add(Treap *t, ll x) {
  t->add += x;
}

void push(Treap *t) {
  t->val += t->add;
  if(t->l)  t->l->add += t->add;
  if(t->r)  t->r->add += t->add;
  t->add = 0;
}

void pull(Treap *t) {
  t->sum = sum(t->l) + sum(t->r) + t->val;
  t->sz = sz(t->l) + sz(t->r) + 1;
}

Treap* merge(Treap *a, Treap *b) {
  if(!a || !b)  return a ? a : b;
  else if(a->pri > b->pri) {
    push(a);
    a->r = merge(a->r, b);
    pull(a);
    return a;
  }
  else {
    push(b);
    b->l = merge(a, b->l);
    pull(b);
    return b;
  }
}

void split(Treap* t, int k, Treap *&a, Treap *&b) {
  if(!t)  a = b = NULL;
  else if(sz(t->l) < k) {
    a = t;
    push(a);
    split(t->r, k - sz(t->l) - 1, a->r, b);
    pull(a);
  }
  else {
    b = t;
    push(b);
    split(t->l, k, a, b->l);
    pull(b);
  }
}

int main() {
  srand(105105);

  int n, q;
  scanf("%d%d", &n, &q);

  Treap *t = NULL;
  for(int i = 0; i < n; i++) {
    ll tmp;
    scanf("%lld", &tmp);
    t = merge(t, make(tmp));
  }

  while(q--) {
    char c;
    int l, r;
    scanf("\n%c %d %d", &c, &l, &r);

    Treap *tl = NULL, *tr = NULL;
    if(c == 'Q') {
      split(t, l - 1, tl, t);
      split(t, r - l + 1, t, tr);
      printf("%lld\n", sum(t));
      t = merge(tl, merge(t, tr));
    }
    else {
      ll x;
      scanf("%lld", &x);
      split(t, l - 1, tl, t);
      split(t, r - l + 1, t, tr);
      add(t, x);
      t = merge(tl, merge(t, tr));
    }
  }

  return 0;
}
```

6.2 copy on write treap

```cpp
#include <cstdlib>
#include <cstdio>
#include <algorithm>
#include <climits>
#include <cstring>

using namespace std;

const int N = 1000000 + 10;

struct Treap {
    char val;
    int sz, refs;
    Treap *l, *r;

    Treap() {}
    Treap(char _val):
        val(_val), sz(1), refs(0), l(NULL),
            r(NULL) {}
};

Treap* make(Treap* t) {
    return new Treap(*t);
}

Treap* make(char _val) {
    return new Treap(_val);
}

void print_ref(Treap* t) {
    if(!t)  return ;
    print_ref(t->l);
    printf("%d ", t->refs);
    print_ref(t->r);
}

void print(Treap* t) {
    if(!t)  return ;
    print(t->l);
    putchar(t->val);
    print(t->r);
}

void takeRef(Treap* t) {
    if(t)   t->refs++;
}

void dropRef(Treap* t) {
    if(t) {
        char c = t->val;
        t->refs--;
        if(t->refs <= 0) {
            dropRef(t->l);
            dropRef(t->r);
            delete t;
        }
    }
}

int sz(Treap* t) {
    return t ? t->sz : 0;
}

int rnd(int m) {
    static int x = 851025;
    return (x = (x*0xdefaced+1) & INT_MAX)
        % m;
}

void pull(Treap* t) {
    t->sz = sz(t->l) + sz(t->r) + 1;
}

Treap* merge(Treap* a, Treap* b) {
    if(!a || !b) {
        Treap* t = a ? make(a) : make(b);
        t->refs = 0;
        takeRef(t->l);
        takeRef(t->r);
        return t;
    }

    Treap* t;
    if( rnd(a->sz+b->sz) < a->sz) {
        t = make(a);
        t->refs = 0;
        t->r = merge(a->r, b);
        takeRef(t->l);
        takeRef(t->r);
    }
    else {
        t = make(b);
        t->refs = 0;
        t->l = merge(a, b->l);
        takeRef(t->l);
        takeRef(t->r);
    }

    pull(t);
    return t;
}

void split(Treap* t, int k, Treap* &a,
    Treap* &b) {
    if(!t)  a = b = NULL;
    else if(sz(t->l) < k) {
        a = make(t);
        a->refs = 0;
        split(a->r, k-sz(t->l)-1, a->r, b);
        takeRef(a->l);
        takeRef(a->r);
        pull(a);
    }
    else {
        b = make(t);
        b->refs = 0;
        split(b->l, k, a, b->l);
        takeRef(b->l);
        takeRef(b->r);
        pull(b);
    }
}

void print_inorder(Treap* t) {
    if(!t)  return ;
    putchar(t->val);
    print_inorder(t->l);
    print_inorder(t->r);
}

```

```
char s[N];

int main() {
    int m;
    scanf("%d", &m);
    scanf("%s", s);
    int n = strlen(s);
    int q;
    scanf("%d", &q);

    Treap* t = NULL;
    for(int i = 0; i < n; i++) {
        Treap *a = t, *b = make(s[i]);
        t = merge(a, b);
        dropRef(a);
        dropRef(b);
    }

    while(q--) {
        int l, r, x;
        scanf("%d%d%d", &l, &r, &x);
        r++;

        Treap *a, *b, *c, *d;
        a = b = c = d = NULL;
        split(t, l, a, b);
        dropRef(a);
        split(b, r-l, c, d);
        dropRef(b);
        dropRef(d);
        split(t, x, a, b);
        dropRef(t);
        Treap* t2 = merge(c, b);
        dropRef(b);
        dropRef(c);
        t = merge(a, t2);
        dropRef(a);
        dropRef(t2);

        if(t->sz > m) {
            Treap* t2 = NULL;
            split(t, m, t2, a);
            dropRef(a);
            dropRef(t);
            t = t2;
        }
    }

    print(t);
    putchar('\n');

    return 0;
}
```

6.3 copy on write segment tree

```
#include <cstdlib>
#include <cstdio>
#include <algorithm>
#include <vector>

using namespace std;

const int N = 50000 + 10;
const int Q = 10000 + 10;

struct Seg {
  static Seg mem[N*80], *pmem;

  int val;
  Seg *tl, *tr;

  Seg() :
    tl(NULL), tr(NULL), val(0) {}

  Seg* init(int l, int r) {
    Seg* t = new (pmem++) Seg();
    if(l != r) {
      int m = (l+r)/2;
      t->tl = init(l, m);
      t->tr = init(m+1, r);
    }
    return t;
  }

  Seg* add(int k, int l, int r) {
    Seg* _t = new (pmem++) Seg(*this);
    if(l==r) {
      _t->val++;
      return _t;
    }

    int m = (l+r)/2;
    if(k <= m)  _t->tl = tl->add(k, l, m);
    else        _t->tr = tr->add(k, m+1, r);

    _t->val = _t->tl->val + _t->tr->val;
    return _t;
  }
} Seg::mem[N*80], *Seg::pmem = mem;

int query(Seg* ta, Seg* tb, int k, int l,
    int r) {
  if(l == r)  return l;

  int m = (l+r)/2;

  int a = ta->tl->val;
  int b = tb->tl->val;
  if(b-a >= k)  return query(ta->tl, tb->tl
      , k, l, m);
  else        return query(ta->tr, tb->tr, k
      -(b-a), m+1, r);
};

struct Query {
  int op, l, r, k, c, v;

  bool operator<(const Query b) const {
    return c < b.c;
  }
} qs[Q];
int arr[N];
Seg *t[N];
vector<int> vec2;

int main() {
  int T;
  scanf(”%d”, &T);

  while(T--) {
    int n, q;
    scanf(”%d%d”, &n, &q);

    for(int i = 1; i <= n; i++) {
      scanf(”%d”, arr+i);
      vec2.push_back(arr[i]);
    }
    for(int i = 0; i < q; i++) {
      scanf(”%d”, &qs[i].op);
      if(qs[i].op == 1) scanf(”%d%d%d”, &qs[i].l, &qs[i].r, &qs[i].k);
      else  scanf(”%d%d”, &qs[i].c, &qs[i].v);

      if(qs[i].op == 2) vec2.push_back(qs[i].v);
    }
    sort(vec2.begin(), vec2.end());
    vec2.resize(unique(vec2.begin(), vec2.end())-vec2.begin());
    for(int i = 1; i <= n; i++) arr[i] = lower_bound(vec2.begin(), vec2.end(), arr[i]) - vec2.begin();
    int mn = 0, mx = vec2.size()-1;

    for(int i = 0; i <= n; i++) t[i] = NULL;
    t[0] = new (Seg::pmem++) Seg();
    t[0] = t[0]->init(mn, mx);
    int ptr = 0;
    for(int i = 1; i <= n; i++) {
      t[i] = t[i-1]->add(arr[i], mn, mx);
    }

    for(int i = 0; i < q; i++) {
      int op = qs[i].op;
      if(op == 1) {
        int l = qs[i].l, r = qs[i].r, k = qs[i].k;
        printf(”%d\n”, vec2[query(t[l-1], t[r], k, mn, mx)]);
      }
      if(op == 2) {
        continue;
      }
      if(op == 3) puts(”7122”);
    }

    vec2.clear();
    Seg::pmem = Seg::mem;
  }

  return 0;
}
```

6.4 Treap+(HOJ 92)

```
#include <cstdlib>
#include <cstdio>
#include <algorithm>
#include <cstring>

using namespace std;

const int INF = 103456789;

struct Treap {
    int pri, sz, val, chg, rev, sum, lsum, rsum, mx_sum;
    Treap *l, *r;

    Treap() {}
    Treap(int _val) :
        pri(rand()), sz(1), val(_val), chg(INF), rev(0), sum(_val), lsum(_val), rsum(_val), mx_sum(_val), l(NULL), r(NULL) {}
};

int sz(Treap* t) {return t ? t->sz : 0;}
int sum(Treap* t) {
    if(!t)  return 0;
    if(t->chg == INF)   return t->sum;
    else    return t->chg*t->sz;
}
int lsum(Treap* t) {
    if(!t)  return -INF;
    if(t->chg != INF)   return max(t->chg, (t->chg)*(t->sz));
    if(t->rev)  return t->rsum;
    return t->lsum;
}
int rsum(Treap* t) {
    if(!t)  return -INF;
    if(t->chg != INF)   return max(t->chg, (t->chg)*(t->sz));
    if(t->rev)  return t->lsum;
    return t->rsum;
}
int mx_sum(Treap* t) {
    if(!t)  return -INF;
    if(t->chg != INF)   return max(t->chg, (t->chg)*(t->sz));
    return t->mx_sum;
}

void push(Treap* t) {
    if(t->chg != INF) {
        t->val = t->chg;
        t->sum = (t->sz) * (t->chg);
        t->lsum = t->rsum = t->mx_sum = max(t->sum, t->val);
        if(t->l)    t->l->chg = t->chg;
        if(t->r)    t->r->chg = t->chg;
        t->chg = INF;
    }
    if(t->rev) {
        swap(t->l, t->r);
        if(t->l)    t->l->rev ^= 1;
        if(t->r)    t->r->rev ^= 1;
        t->rev = 0;
    }
}

void pull(Treap* t) {
    t->sz = sz(t->l)+sz(t->r)+1;
    t->sum = sum(t->l)+sum(t->r)+t->val;
    t->lsum = max(lsum(t->l), sum(t->l)+max
        (0, lsum(t->r))+t->val);
    t->rsum = max(rsum(t->r), sum(t->r)+max
        (0, rsum(t->l))+t->val);
    t->mx_sum = max(max(mx_sum(t->l),
        mx_sum(t->r)), max(0, rsum(t->l))+
        max(0, lsum(t->r))+t->val);
}

Treap* merge(Treap* a, Treap* b) {
    if(!a || !b)    return a ? a : b;
    if(a->pri > b->pri) {
        push(a);
        a->r = merge(a->r, b);
        pull(a);
        return a;
    }
    else {
        push(b);
        b->l = merge(a, b->l);
        pull(b);
        return b;
    }
}

void split(Treap* t, int k, Treap* &a,
    Treap* &b) {
    if(!t) {
        a = b = NULL;
        return ;
    }
    push(t);
    if(sz(t->l) < k) {
        a = t;
        push(a);
        split(t->r, k-sz(t->l)-1, a->r, b);
        pull(a);
    }
    else {
        b = t;
        push(b);
        split(t->l, k, a, b->l);
        pull(b);
    }
}

void del(Treap* t) {
    if(!t)  return;
    del(t->l);
    del(t->r);
    delete t;
}

int main() {
    srand(7122);

    int n, m;
    scanf("%d%d", &n, &m);

    Treap* t = NULL;
    for(int i = 0; i < n; i++) {
        int x;
        scanf("%d", &x);
        t = merge(t, new Treap(x));
    }

    while(m--) {
        char s[15];
        scanf("%s", s);

        Treap *tl = NULL, *tr = NULL, *t2 =
            NULL;

        if(!strcmp(s, "INSERT")) {
            int p, k;
            scanf("%d%d", &p, &k);
            for(int i = 0; i < k; i++) {
                int x;
                scanf("%d", &x);
                t2 = merge(t2, new Treap(x)
                    );
            }
            split(t, p, tl, tr);
            t = merge(tl, merge(t2, tr));
        }

        if(!strcmp(s, "DELETE")) {
            int p, k;
            scanf("%d%d", &p, &k);
            split(t, p-1, tl, t);
            split(t, k, t, tr);
            del(t);
            t = merge(tl, tr);
        }

        if(!strcmp(s, "MAKE-SAME")) {
            int p, k, l;
            scanf("%d%d%d", &p, &k, &l);
            split(t, p-1, tl, t);
            split(t, k, t, tr);
            if(t)   t->chg = l;
            t = merge(tl, merge(t, tr));
        }

        if(!strcmp(s, "REVERSE")) {
            int p, k;
            scanf("%d%d", &p, &k);
            split(t, p-1, tl, t);
            split(t, k, t, tr);
            if(t)   t->rev ^= 1;
            t = merge(tl, merge(t, tr));
        }

        if(!strcmp(s, "GET-SUM")) {
            int p, k;
            scanf("%d%d", &p, &k);
            split(t, p-1, tl, t);
            split(t, k, t, tr);
            printf("%d\n", sum(t));
            t = merge(tl, merge(t, tr));
        }

        if(!strcmp(s, "MAX-SUM")) {
            printf("%d\n", mx_sum(t));
        }
    }

    return 0;
```

```
}
```

6.5 Leftist Tree

```
#include <bits/stdc++.h>
using namespace std;

struct Left{
  Left *l,*r;
  int v,h;
  Left(int v_) : v(v_), h(1), l(0), r(0) {}
};

int height(Left *p)
{
  return p ? p -> h : 0 ;
}

Left* combine(Left *a,Left *b)
{
  if(!a || !b) return a ? a : b ;
  Left *p ;
  if( a->v > b->v)
  {
    p = a;
    p -> r = combine( p -> r , b );
  }
  else
  {
    p = b;
    p -> r = combine( p -> r , a );
  }
  if( height( p->l ) < height( p->r ) )
    swap( p->l , p->r );
  p->h = min( height( p->l ) , height( p->r ) ) + 1;
  return p;
}
Left *root;

void push(int v)
{
  //printf("push-%d\n",v);
  Left *p = new Left(v);
  root = combine( root , p );
  //puts("end");
}
int top()
{
  return root? root->v : -1;
}
void pop()
{
  if(!root) return;
  Left *a = root->l , *b = root->r ;
  delete root;
  root = combine( a , b );
}
void clear(Left* &p)
{
  if(!p)
    return;
  if(p->l) clear(p->l);
  if(p->r) clear(p->r);
  delete p;
  p = 0 ;
}


int main()
{
  int T,n,x,o,size;
  bool bst,bqu,bpq;
  scanf("%d",&T);
  while(T--)
  {
    bst=bqu=bpq=1;
    stack<int> st;
    queue<int> qu;
    clear(root);
    size=0;
    scanf("%d",&n);
    while(n--)
    {
      scanf("%d%d",&o,&x);
      if(o==1)
        st.push(x),qu.push(x),push(x),size++;
      else if(o==2)
      {
        size--;
        if(size<0)
          bst=bqu=bpq=0;
        if(bst)
        {
          if(st.top()!=x)
            bst=0;
          st.pop();
        }
        if(bqu)
        {
          if(qu.front()!=x)
            bqu=0;
          qu.pop();
        }
        if(bpq)
        {
        //  printf("(%d)\n",top());
          if(top()!=x)
            bpq=0;
          pop();
        }
      }
    }
    int count=0;
    if(bst)
      count++;
    if(bqu)
      count++;
    if(bpq)
      count++;

    if(count>1)
      puts("not sure");
    else if(count==0)
      puts("impossible");
    else if(bst)
      puts("stack");
    else if(bqu)
      puts(”queue”);
    else if(bpq)
      puts(”priority queue”);
  }
  return 0;
}
```

7 geometry

7.1 Basic

```
// correct code of NPSC2013 senior-final pF

#include <bits/stdc++.h>
#define pb push_back
#define F first
#define S second
#define SZ(x) ((int)(x).size())
#define MP make_pair
using namespace std;
typedef long long ll;
typedef pair<int,int> PII;
typedef vector<int> VI;

typedef double dou;
struct PT{
  dou x,y;
  PT(dou x_=0.0,dou y_=0.0): x(x_),y(y_) {}
  PT operator + (const PT &b) const { return PT(x+b.x,y+b.y); }
  PT operator - (const PT &b) const { return PT(x-b.x,y-b.y); }
  PT operator * (const dou &t) const { return PT(x*t,y*t); }
  dou operator * (const PT &b) const { return x*b.x+y*b.y; }
  dou operator % (const PT &b) const { return x*b.y-b.x*y; }
  dou len2() const { return x*x+y*y; }
  dou len() const { return sqrt(len2()); }
};

const dou INF=1e12;
const dou eps=1e-8;
PT inter(const PT &P1,const PT &T1,const PT &P2,const PT &T2) // intersection
{
  if(fabs(T1%T2)<eps)
    return PT(INF,INF);
  dou u=((P2-P1)%T2)/(T1%T2);
  return P1+T1*u;
}

PT conv[500],cat,to;

int main()
{
  int T,N,M;
  scanf(”%d”,&T);
  while(T--)
  {
    scanf(”%d%d”,&N,&M);
    for(int i=0;i<N;i++)
      scanf(”%lf%lf”,&conv[i].x,&conv[i].y);
    conv[N]=conv[0];
    dou ans=0.0;
    while(M--)
    {
      scanf(”%lf%lf%lf%lf”,&cat.x,&cat.y,&to.x,&to.y);
      for(int i=0;i<N;i++)
        if(fabs((conv[i]-conv[i+1])%to)>eps)
        {
        //  printf(”M:%d i=%d\n”,M,i);
          PT at=inter(conv[i],conv[i]-conv[i+1],cat,to);
          if((conv[i]-at)*(conv[i+1]-at)<eps && (at-cat)*to>-eps)
            ans=max(ans,(cat-at).len());
        }
    }
    printf(”%.4f\n”,ans);
  }
  return 0;
}
```

7.2 Smallist circle problem

```
#include <cstdlib>
#include <cstdio>
#include <algorithm>
#include <cmath>

//#define test

using namespace std;

const int N = 1000000 + 10;

struct PT {
  double x, y;

  PT() {}
  PT(double x, double y):
    x(x), y(y) {}
  PT operator+(const PT &b) const {
    return (PT) {x+b.x, y+b.y};
  }
  PT operator-(const PT &b) const {
    return (PT) {x-b.x, y-b.y};
  }
  PT operator*(const double b) const {
    return (PT) {x*b, y*b};
  }
  PT operator/(const double b) const {
    return (PT) {x/b, y/b};
  }
  double operator%(const PT &b) const {
    return x*b.y - y*b.x;
  }

  double len() const {
    return sqrt(x*x + y*y);
  }
  PT T() const {
    return (PT) {-y, x};
  }
} p[N];

void update(PT a, PT b, PT c, PT &o, double &r) {
  if(c.x < 0.0) o = (a+b) / 2.0;
  else {
    PT p1 = (a+b)/2.0, p2 = p1 + (b-a).T();
    PT p3 = (a+c)/2.0, p4 = p3 + (c-a).T();
    double a123 = (p2-p1)%(p3-p1), a124 = (p2-p1)%(p4-p1);
    if(a123 * a124 > 0.0) a123 = -a123;
    else  a123 = abs(a123), a124 = abs(a124);
    o = (p4*a123 + p3*a124) / (a123 + a124);
  }
  r = (a-o).len();
}

int main() {
  //freopen(”C:/Users/S11/Desktop/pb.in”, ”r”, stdin);

  srand(7122);

  int m, n;
  while(scanf(”%d%d”, &m, &n)) {
    if(!n && !m)  return 0;

    for(int i = 0; i < n; i++)  scanf(”%lf%lf”, &p[i].x, &p[i].y);

    for(int i = 0; i < n; i++)
      swap(p[i], p[rand() % (i+1)]);

    PT a = p[0], b = p[1], c(-1.0, -1.0), o = (a+b) / 2.0;
    double r = (a-o).len();
    for(int i = 2; i < n; i++) {
      if((p[i]-o).len() <= r) continue;

      a = p[i];
      b = p[0];
      c = (PT) {-1.0, -1.0};
      update(a, b, c, o, r);
      for(int j = 1; j < i; j++) {
        if((p[j]-o).len() <= r) continue;

        b = p[j];
        c = (PT) {-1.0, -1.0};
        update(a, b, c, o, r);

        for(int k = 0; k < j; k++) {
          if((p[k]-o).len() <= r) continue;

          c = p[k];
          update(a, b, c, o, r);
        }
      }

      #ifdef test
      printf(”i=%d\n”, i);
      printf(”a=(%.1f, %.1f)\n”, a.x, a.y);
      printf(”b=(%.1f, %.1f)\n”, b.x, b.y);
      printf(”c=(%.1f, %.1f)\n”, c.x, c.y);
      printf(”o=(%.1f, %.1f)\n”, o.x, o.y);
      printf(”r=%.1f\n”, r);
      puts(”-----”);
      #endif // test
    }

    printf(”%.3f\n”, r);
  }
}
```