南京大学 ACM-ICPC 集训队
calabash_boy
代码模版库

目录

1 String

1.1 Hash

```
427e // Created by calabash_boy on 18-6-1.
427e // CF 1003F
302f #include<bits/stdc++.h>
421c using namespace std;
b773 typedef unsigned long long ULL;
93c3 const int maxn = 305*305;
75c0 /* 字符集大小 */
0852 const int sigma = maxn;
0338 /* hash次数 */
cab3 const int HASH_CNT = 2;
5c83 int n;
4c95 int s[maxn];
bef3  /* char* 1-bas
5cb4  * sum[i] = s[i]+s[i-1]*Seed+s[i-2]*Seed^2+...+s[1]*Seed^(i-1)*/
cf6f ULL Prime_Pool[] = {1998585857ul,23333333333ul};
d095 ULL Seed_Pool[]={911,146527,19260817,91815541};
c437 ULL Mod_Pool[]={29123,998244353,1000000009,4294967291ull};
b060 struct Hash_1D{
3e0c     ULL Seed,Mod;
3bc4     ULL bas[maxn];ULL sum[maxn];
ad94     int perm[sigma];
be03     void init(int seedIndex,int modIndex){
e7a7         Seed = Seed_Pool[seedIndex];
53c7         Mod = Mod_Pool[modIndex];
bf6d         bas[0]=1;
6dbf         for (int i=1;i<=n;i++){
d57c             bas[i] = bas[i-1]*Seed%Mod;
95cf         }
6dbf         for (int i=1;i<=n;i++){
1e15             sum[i] = (sum[i-1]*Seed%Mod+s[i])%Mod;
95cf         }
95cf     }
c2c1     /*random_shuffle 离散化id，防止kill_hash*/
b864     void indexInit(int seedIndex,int modIndex){
324a         for (int i=1;i<n;i++){
871a             perm[i]=i;
95cf         }
cee0         random_shuffle(perm+1,perm+1+sigma);
e7a7         Seed = Seed_Pool[seedIndex];
53c7         Mod = Mod_Pool[modIndex];
bf6d         bas[0]=1;
6dbf         for (int i=1;i<=n;i++){
d57c             bas[i] = bas[i-1]*Seed%Mod;
95cf         }
6dbf         for (int i=1;i<=n;i++){
cd52             sum[i] = (sum[i-1]*Seed%Mod+perm[s[i]])%Mod;
95cf         }
95cf     }
b2c3     ULL getHash(int l,int r){
46bc         return (sum[r]-sum[l-1]*bas[r-l+1]%Mod+Mod)%Mod;
95cf     }
bb59 }hasher[HASH_CNT];
f09b map<pair<pair<ULL,ULL>,int>,int>veid;int vecnt;
5d53 map<string,int>id;int idcnt;
7fbd vector<int> pos[maxn];
fae2 string a[maxn];
f06b int sumL[maxn];
3117 int main(){
e1b6     cin>>n;
6dbf     for (int i=1;i<=n;i++){
879c         cin>>a[i];
d0a8         if (!id[a[i]])id[a[i]] = ++idcnt;
7798         s[i] = id[a[i]];
9892         sumL[i] = sumL[i-1]+a[i].size();
95cf     }
da02     for (int i=0;i<HASH_CNT;i++){
42fc         hasher[i].indexInit(i,i);
95cf     }
b20c     int ans = sumL[n]+n-1;
6dbf     for (int i=1;i<=n;i++){
ede7         for (int j=1;j<=n;j++){
e9bb             ULL hash1 = hasher[0].getHash(i,j);
2a70             ULL hash2 = hasher[1].getHash(i,j);
de4a             int len = j-i+1;
46fa             pair<pair<ULL,ULL>,int> x = {{hash1,hash2},len};
67ca             if (veid[x]==0)veid[x] = ++vecnt;
2251             pos[veid[x]].push_back(i);
95cf         }
95cf     }
04c1     int maxDelta =0;
0086     for (auto x:veid){
5c1e         int len = x.first.second;
76c1         int i = x.second;
```

```
3492        sort(pos[i].begin(),pos[i].end());
978f        int num =0;
6866        for (int j=0,last = -maxn;j<pos[i].size();j++){
683e            if (pos[i][j]>=last+len){
56e2                last = pos[i][j];
ac46                num++;
95cf            }
95cf        }
162f        if (num==1)continue;
e8b3        int cost1 = sumL[pos[i][0]+len-1]-sumL[pos[i][0]-1]+len-1;
939d        int cost2 = len;
5770        int tempDelta = (cost1-cost2)*num;
7f18        maxDelta = max(maxDelta,tempDelta);
95cf    }
cce6    cout<<ans-maxDelta<<endl;
7021    return 0;
95cf}
```

1.2 KMP

```
427e// Created by calabash_boy on 18-7-23.
427e//最小权值和 二维循环节
427e//找到最小 每行公共循环节+每列公共循环节。
427e//单调队列找固定大小矩形最小权值和。
302f#include<bits/stdc++.h>
421cusing namespace std;
94a1const int maxn = 1e6+100;
a239struct KMP{
8323    int nxt[maxn];int len;
0409    char t[maxn];
1126    void clear(){
3c88        len =nxt[0] = nxt[1] =0;
95cf    }
c0bf    /* 1-bas */
b115    /* 注意在ss结尾添加 '\0' */
2e3f    void init(char* ss){
64a4        len = strlen(ss+1);
b596        memcpy(t,ss,(len+2)*sizeof(char));
ca76        for (int i=2;i<=len;i++){
362a            nxt[i] = nxt[i-1];
bbb0            while (nxt[i]&&ss[i]!=ss[nxt[i]+1]) nxt[i] = nxt[nxt[i]];
da9f            nxt[i]+=(ss[i]==ss[nxt[i]+1]);
95cf        }
95cf    }
95cf}
49e0/* 求所有在ss串中的start_pos. 如果first_only设置为true，则只返回第一个位置*/
5cfdvector<int> match(char *ss,bool first_only = false){
55fa    int len_s = strlen(ss+1);
8364    vector<int> start_pos(0);
d287    for (int i=1,j=1;i<=len_s;){
8957        while (j!=1 && ss[i] != t[j])j = nxt[j-1]+1;
517c        if (ss[i] == t[j]) j++,i++;
aabb        else i++;
ffa2        if (j == len+1){
741d            start_pos.push_back(i-j+1);
f056            if (first_only)return start_pos;
d0e6            j = nxt[len]+1;
95cf        }
95cf    }
17e3    return start_pos;
95cf}
56ddvoid debug(){
0d69    for (int i=0;i<=len;i++){
3cb0        printf("[debug] nxt[%d]=%d\n",i,nxt[i]);
95cf    }
95cf}
243b/* 循环周期 形如 acaca 中 ac 是一个合法周期 */
d4e9vector<int> periodic(){
995a    vector<int> ret;
4a5d    int now = len;
3f78    while (now){
ebeb        now = nxt[now];
9341        ret.push_back(len-now);
95cf    }
ee0f    return ret;
95cf}
f525/* 循环节 形如 acac 中ac、acac是循环节，aca不是*/
1a85vector<int> periodic_loop(){
995a    vector<int>ret ;
d561    for (int x :periodic()){
901d        if (len%x==0)ret.push_back(x);
95cf    }
ee0f    return ret;
95cf}
5531int min_periodic_loop(){
8b2c    return periodic_loop()[0];
95cf}
```

```
}kmper;
vector<string> s;
vector<vector<int> > a,maxVal;
int cnt1[maxn],cnt2[maxn],n,m;
char S[maxn];
pair<int,int> pq[maxn];int l,r;
int main(){
    cin>>n>>m;
    s.resize(n+1);
    maxVal.resize(n+1);
    for (int i=1; i<=n;i++){
        cin>>s[i];
    }
    a.resize(n+1);
    for (int i=1;i<=n;i++){
        a[i].resize(m+1);
        maxVal[i].resize(m+1);
        for (int j=1;j<=m;j++){
            cin>>a[i][j];
        }
    }
    int p,q;kmper.clear();
    for (int i=1;i<=n;i++){
        for (int j=1;j<=m;j++){
            S[j] = s[i][j-1];
        }
        S[m+1]='\0';
        kmper.init(S);
        for (int x:kmper.periodic()){
            cnt1[x]++;
        }
    }
    for (int j=1;j<=m;j++){
        for (int i=1;i<=n;i++){
            S[i] = s[i][j-1];
        }
        S[n+1]='\0';
        kmper.init(S);
        for (int x:kmper.periodic()){
            cnt2[x]++;
        }
    }
    for (int i=maxn;i>=1;i--){
        if (cnt1[i]==n){ q = i; }
        if (cnt2[i]==m){ p=i; }
    }
    for (int i=1;i<=n;i++){
        l = 0,r=0;
        for (int j=1;j<=m;j++){
            while (r>l&&pq[l].second<=j-q)l++;
            while (r>l&&pq[r-1].first<=a[i][j])r--;
            pq[r++] = {a[i][j],j};
            if (j>=q){
                maxVal[i][j-q+1] = pq[l].first;
            }
        }
    }
    int ans = 0x3f3f3f3f;
    for (int j=1;j<=m-q+1;j++){
        l=r=0;
        for (int i=1;i<=n;i++){
            while (r>l&&pq[l].second<=i-p)l++;
            while (r>l&&pq[r-1].first<=maxVal[i][j])r--;
            pq[r++] = {maxVal[i][j],i};
            if (i>=p){
                ans = min(ans,pq[l].first);
            }

        }
    }
    cout<<1LL*(p+1)*(q+1)*ans<<endl;
    return 0;
}
```

1.3 Manacher

```
// Created by calabash_boy on 18-9-14.
#include<bits/stdc++.h>
using namespace std;
const int MAX = 2e5+10000;
char s[MAX];
struct Manacher{
    int lc[MAX];
    char ch[MAX];
    int N;
    Manacher(char *s){init(s);manacher();}
```

```
44ca     /* s 1 bas */
e798     void init(char *s){
0de8         int n = strlen(s+1);
ad19         ch[n*2 +1] = '#';
ce0d         ch[0] = '@';
46cd         ch[n*2 +2] = '\0';
0c3f         for (int i=n;i>=1;i--){
6beb             ch[i*2] = s[i];ch[i*2 -1] = '#';
95cf         }
5991         N = 2* n +1;
95cf     }
6c5f     void manacher(){
a461         lc[1]=1;  int k=1;
256b         for (int i=2;i<=N;i++){
7957             int p = k+lc[k]-1;
5e04             if (i<=p){
24a1                 lc[i]=min(lc[2*k-i],p-i+1);
87d6             }else{  lc[i]=1;  }
aa80             while (ch[i+lc[i]]==ch[i-lc[i]])lc[i]++;
2b9a             if (i+lc[i]>k+lc[k])k=i;
95cf         }
95cf     }
56dd     void debug(){
b492         puts(ch);
cd0f         for (int i=1;i<=N;i++){
0d62             printf("lc[%d]=%d\n",i,lc[i]);
95cf         }
95cf     }
329b };
3117 int main(){
a275     scanf("%s",s+1);
382e     Manacher manacher(s);
9c07     manacher.debug();
7021     return 0;
95cf }
```

1.4 Suffix Array

```
87e7 /*
1e1d  * for each 2-power string.
f606  * let its length is 2L. add edge of length w[L] between every i and i + L.
f3db  * calculate the spanning forests.
 */                                                                          f2b5
#include <bits/stdc++.h>                                                     302f
#define rank rkrkrk                                                          18f5
//#define _DEBUG                                                             427e
#define RMQ                                                                  f11b
using namespace std;                                                         421c
const int maxn = 3e5+100;                                                    6428
int w[maxn];                                                                 82ea
struct Run{                                                                  2f33
    int l,r,k;                                                               8f36
};                                                                           329b
struct UFS {                                                                 bd89
    int fa[maxn];                                                            33ef
    void init(int n) { iota(fa, fa + n + 1, 0); }                            7dd9
    int find(int x) { return fa[x] == x ? x : fa[x] = find(fa[x]); }          38dd
    bool unite(int u, int v) {                                               9662
        u = find(u); v = find(v);                                            576f
        fa[u] = v;                                                           2448
        return u != v;                                                       4042
    }                                                                        95cf
} ufs[20];                                                                   d71b
                                                                             427e
int unite(int u, int v, int k) {                                             4d49
    if (ufs[k].unite(u, v)) {                                                10fe
        if (k == 0) return 1;                                                d11e
        return unite(u, v, k - 1) + unite(u + (1<<(k-1)), v + (1<<(k-1)), k - 1)  81a9
          ;
    } else return 0;                                                         aad3
}                                                                            95cf
                                                                             427e
long long merge(int u, int v, int l) {                                       6b2b
    int k = log2(l);                                                         0fa9
    int ret =  unite(u, v, k) +                                              2c46
               unite(u + l - (1<<k), v + l - (1<<k), k);                     270b
    return ret;                                                              ee0f
}                                                                            95cf
struct SA{                                                                   3b88
#ifndef RMQ                                                                  4eb6
    struct Segment_Tree{                                                     9c29
        int min_val[maxn*4];                                                 77b7
        void up(int x){                                                      d08d
            min_val[x] = min(min_val[x<<1],min_val[x<<1|1]);                 10d7
        }                                                                    95cf
        void build(int x,int l,int r,int*h){                                 3e01
```

```
3a0d            if (l == r){
e948                min_val[x] = h[l];
4f2d                return;
95cf            }
b8b7            int mid = l + r >>1;
fdb0            build(x<<1,l,mid,h);
06e9            build(x<<1|1,mid+1,r,h);
cf00            up(x);
95cf        }
30b1        int query(int x,int l,int r,int L,int R){
133b            if (l > R || L > r)return 0x3f3f3f3f;
0739            if (L<= l && r <= R)return min_val[x];
b8b7            int mid = l + r >> 1;
edf8            return min(query(x<<1,l,mid,L,R),query(x<<1|1,mid+1,r,L,R));
95cf        }
f7fb    }segtree;
a8cb#else
fb7f    int st[maxn][20];
a66e    void st_init(int n,int*h){
6dbf        for (int i=1;i<=n;i++){
fc74            st[i][0] = h[i];
95cf        }
c8a2        for (int j=1;(1<<j)<=n;j++){
672f            for (int i=1;i<=n-(1<<j)+1;i++){
3c6e                st[i][j] = min(st[i][j-1],st[i+(1<<(j-1))][j-1]);
95cf            }
95cf        }
95cf    }
1937#endif
6e4f    int cntA[maxn],cntB[maxn],tsa[maxn],A[maxn],B[maxn];
f3d8    int sa[maxn],rank[maxn],height[maxn];
81e4    void get_sa(int * ch,int n){
b5cc        ch[0] = ch[n+1] = -1;
c7f9        for (int i=0;i<=n;i++)cntA[i] = 0;
e86b        for (int i=1;i<=n;i++)cntA[ch[i]]++;
c35a        for (int i=1;i<=n;i++)cntA[i] += cntA[i-1];
625e        for (int i=n;i>=1;i--)sa[cntA[ch[i]]--] = i;
c9f2        rank[sa[1]] = 1;
a5c5        for (int i=2;i<=n;i++){
dc5c            rank[sa[i]] = rank[sa[i-1]];
459c            if (ch[sa[i]] != ch[sa[i-1]])rank[sa[i]] ++;
95cf        }
f62b        for (int l=1;rank[sa[n]]<n;l<<=1){
c794            for (int i=0;i<=n;i++)cntA[i] = cntB[i] = 0;
6dbf            for (int i=1;i<=n;i++){
d9ab                cntA[A[i] = rank[i]] ++;
c846                cntB[B[i]=(i+l<=n)?rank[i+l]:0]++;
95cf            }
72d7            for (int i=1;i<=n;i++)cntB[i] += cntB[i-1];
4c62            for (int i=n;i>=1;i--)tsa[cntB[B[i]]--] = i;
c35a            for (int i=1;i<=n;i++)cntA[i] += cntA[i-1];
1626            for (int i=n;i>=1;i--)sa[cntA[A[tsa[i]]]--] = tsa[i];
c9f2            rank[sa[1]] = 1;
a5c5            for (int i=2;i<=n;i++){
dc5c                rank[sa[i]] = rank[sa[i-1]];
021c                if (A[sa[i]] != A[sa[i-1]] || B[sa[i]] != B[sa[i-1]])rank[sa[i]]
                        ++;
95cf            }
95cf        }
95cf    }
bbe8    void get_height(int *ch,int n){
0820        get_sa(ch,n);
5c18        sa[0] = rank[0] = 0;
0956        for (int i=1,j=0;i<=n;i++){
1a82            if (j) j--;
757e            while (ch[i+j] == ch[sa[rank[i]-1]+j])j++;
24a7            height[rank[i]] = j;
95cf        }
ed5c#ifdef _DEBUG
6dbf        for (int i=1;i<=n;i++){
dfcf            printf("height[%d]=%d\n",i,height[i]);
95cf        }
1937#endif
4eb6#ifndef RMQ
3b40        segtree.build(1,1,n,height);
a8cb#else
a852        st_init(n,height);
1937#endif
95cf    }
ead2    int get_lcp(int x,int y,int n){
6606        int rkx = rank[x];
a728        int rky = rank[y];
4e5e        if (rkx>rky)swap(rkx,rky);
216a        rkx++;
4eb6#ifndef RMQ
dee6        int lcp = segtree.query(1,1,n,rkx,rky);
a8cb#else
b6ec        int k = log2(rky - rkx+1);
```

```
f5b5         int lcp = min(st[rkx][k],st[rky - (1<<k)+1][k]);
1937 #endif
427e
ed5c #ifdef _DEBUG
33df         printf("[get_lcp] x=%d y=%d rkx=%d,rky=%d,lcp =%d\n",x,y,rkx,rky,lcp);
1937 #endif
9a6a         return lcp;
95cf     }
5a1e }sa1,sa2;
96d9 int ch2[maxn];
4d50 vector<Run> get_run(int*ch,int n){
7c77     sa1.get_height(ch,n);
842e     for (int i=0;i<=n+1;i++){
13b4         ch2[i] = ch[i];
95cf     }
7db6     reverse(ch2+1,ch2+1+n);
945d     sa2.get_height(ch2,n);
c4b1     vector<Run> result(0);
a2dc     int len_max = n/2;
dbca     for (int len = 1;len <=len_max;len ++){
427e         //get_len_run
870e         for (int i=1;i<=n;i+=len){
d3da             int j = i+len;
dd33             if (j >n)break;
f2a5             int lcp = sa1.get_lcp(i,j,n);
8ef0             int lcs = sa2.get_lcp(n+1-i,n+1-j,n);
f20d             lcp = min(lcp,len);
97fa             lcs = min(lcs,len);
2cd9             assert(j+lcp-1<=n);
6a34             assert(i-lcs+1>=1);
ed5c #ifdef _DEBUG
8dbc             printf("i=%d,j=%d,len=%d,lcp=%d,lcs=%d\n",i,j,len,lcp,lcs);
1937 #endif
37d6             if (lcp + lcs - 1 < len)continue;
09d8             int L = j-lcs+1;
856e             int R = j + lcp -1;
ab80             result.push_back((Run){L,R,len});
95cf         }
95cf     }
ed5c #ifdef _DEBUG
7d48     for (Run run : result){
7252         printf("[run]: l=%d, r=%d,k=%d\n",run.l,run.r,run.k);
95cf     }
1937 #endif
```

```
56b0     return result;
95cf }
5c83 int n;
4085 typedef long long ll;
aec3 ll spanning_forest(vector<Run> &runs){
4f70     sort(runs.begin(),runs.end(),[](Run x,Run y){
b6e2         return w[x.k] < w[y.k];
b251     });
19f3     ll ans = 0;
ec84     for (auto& R : runs) {
de4b         int l = R.l, r = R.r;
bbac         ans += 1ll * merge(l - R.k, l, r - l + 1) * w[R.k];
95cf     }
4206     return ans;
95cf }
7767 int ch[maxn];
3117 int main(){
9523     int T;
1fd9     scanf("%d",&T);
60ca     while (T--){
cd91         scanf("%d",&n);
4721         for (int i = 0; i < 20 ; i++) ufs[i].init(n);
d15f         ch[n+1] = -1;
d442         ch[0] = -1;
6dbf         for (int i=1;i<=n;i++){
b3d6             scanf("%d",ch+i);
95cf         }
9f8e         int m = n/2;
e052         for (int i=1;i<=m;i++){
ef59             scanf("%d",w+i);
95cf         }
3690         vector<Run> all_run = get_run(ch,n);
1ccd         printf("%lld\n",spanning_forest(all_run));
95cf     }
7021     return 0;
95cf }
```

2 String Automaton

2.1 ACAM

```
427e // Created by calabash_boy on 18-6-5.
```

```
427e // HDU 6138
427e //给定若干字典串。
427e // query:strx stry 求最长的p,p为strx、stry子串，且p为某字典串的前缀
302f #include<bits/stdc++.h>
421c using namespace std;
52c1 const int maxn = 1e5+100;
6b3e struct Aho_Corasick_Automaton{
427e     //basic
141b     int nxt[maxn*10][26],fail[maxn*10];
7a04     int root,tot;
427e     //special
8f42     int flag[maxn*10];
d3a5     int len[maxn*10];
1126     void clear(){
21a1         memset(nxt[0],0,sizeof nxt[0]);
0ae1         root = tot=0;
95cf     }
ee91     int newnode(){
71cf         tot++;
87f4         memset(nxt[tot],0,sizeof nxt[tot]);
a231         flag[tot] = len[tot]=0;
91fb         return tot;
95cf     }
9bb4     void insert(char *s ){
8f56         int now = root;
f205         while (*s){
e37a             int id = *s-'a';
ce8f             if(!nxt[now][id])nxt[now][id] = newnode();
7134             len[nxt[now][id]] = len[now]+1;
6f00             now = nxt[now][id];
95cf         }
95cf     }
bcf9     void insert(string str){
8f56         int now = root;
10ad         for (int i=0;i<str.size();i++){
25da             int id = str[i]-'a';
ce8f             if(!nxt[now][id])nxt[now][id] = newnode();
7134             len[nxt[now][id]] = len[now]+1;
6f00             now = nxt[now][id];
95cf         }
95cf     }
2114     void build(){
30ee         fail[root] = root;
c19d         queue<int>Q;Q.push(root);
11e5         while (!Q.empty()){
ff8a             int head = Q.front();Q.pop();
414f             for (int i=0;i<26;i++){
c591                 if(!nxt[head][i])continue;
762f                 int temp = nxt[head][i];
c509                 fail[temp] = fail[head];
a7fb                 while (fail[temp]&&!nxt[fail[temp]][i]){
5e80                     fail[temp] = fail[fail[temp]];
95cf                 }
3198                 if(head&&nxt[fail[temp]][i])fail[temp] = nxt[fail[temp]][i];
6b09                 Q.push(temp);
95cf             }
95cf         }
95cf     }
fddd     void search(string str,int QID);
cf07     int query(string str,int QID);
5ede }acam;
1874 void Aho_Corasick_Automaton::search(string str,int QID) {
8f56     int now = root;
10ad     for (int i=0;i<str.size();i++){
25da         int id = str[i]-'a';
b2b6         now = nxt[now][id];int temp = now;
694e         while (temp!=root&&flag[temp]!=QID){
22a4             flag[temp] = QID;
f597             temp = fail[temp];
95cf         }
95cf     }
95cf }
126b int Aho_Corasick_Automaton::query(string str, int QID) {
81f4     int ans =0;int now = root;
10ad     for (int i=0;i<str.size();i++){
25da         int id = str[i]-'a';
6f00         now = nxt[now][id];
c20a         int temp = now;
dead         while (temp!=root){
497d             if(flag[temp]==QID){
79cd                 ans = max(ans,len[temp]);
6173                 break;
95cf             }
f597             temp = fail[temp];
95cf         }
95cf     }
4206     return ans;
95cf }
```

```
fae2 string a[maxn];
24df int m,n,qid;
3117 int main(){
42db     int T;cin>>T;
60ca     while (T--){
67f3         acam.clear();cin>>n;
6dbf         for (int i=1;i<=n;i++){
879c             cin>>a[i];
e321             acam.insert(a[i]);
95cf         }
1ccd         acam.build();cin>>m;
e052         for (int i=1;i<=m;i++){
74ca             int x,y;cin>>x>>y;
6a4f             qid++;
071c             acam.search(a[x],qid);
c2f3             int ans = acam.query(a[y],qid);
d592             cout<<ans<<endl;
95cf         }
95cf     }
7021     return 0;
95cf }
```

2.2 SAM

```
427e // Created by calabash_boy on 18-6-4.
427e //SPOJ substring
427e // calc ans_i=长度=i的所有子串，出现次数最多的一种出现了多少次。
302f #include<bits/stdc++.h>
374e #define RIGHT
427e //RIGHT: parent树的dfs序上主席树，求每个点的Right集合
421c using namespace std;
40fb const int maxn = 25e4+100;
d273 struct Node{int L,R,val;}Tree[maxn*40];
dd0f #ifdef RIGHT
6207 struct Chairman_Tree{
8abb     int cnt = 0;
bd4f     int root[maxn*2];
5d53     void init(){
a4f5         memset(root,0,sizeof root);
8766         cnt =0;
95cf     }
94cf     /* 建T0空树 */
cf84     int buildT0(int l, int r){
64f2         int k = cnt++;
ecaf         Tree[k].val =0;
eb40         if (l==r) return k;
b8b7         int mid = l+r >>1;
0bf4         Tree[k].L = buildT0(l, mid);Tree[k].R = buildT0(mid + 1, r);
e27b         return k;
95cf     }
e965     /* 上一个版本节点P，【ppos】+=del 返回新版本节点*/
3a6b     int update (int P,int l,int r,int ppos,int del){
d4b1         assert(cnt < maxn*50);
64f2         int k = cnt++;
73d2         Tree[k].val = Tree[P].val +del;
eb40         if (l==r) return k;
b8b7         int mid = l+r >>1;
4af7         if (ppos<=mid){
5b36             Tree[k].L = update(Tree[P].L,l,mid,ppos,del);
de01             Tree[k].R = Tree[P].R;
8e2e         }else{
0d44             Tree[k].L = Tree[P].L;
a179             Tree[k].R = update(Tree[P].R,mid+1,r,ppos,del);
95cf         }
e27b         return k;
95cf     }
b13a     int query(int PL,int PR,int l,int r,int L,int R){
b8e7         if (l>R || L>r)return 0;
03d9         if (L <= l && r <= R)return Tree[PR].val - Tree[PL].val;
b8b7         int mid = l + r >> 1;
ff4f         return query(Tree[PL].L,Tree[PR].L,l,mid,L,R) + query(Tree[PL].R,Tree[PR
               ].R,mid+1,r,L,R);
95cf     }
b0c1 }tree;
1937 #endif
6f83 char s[maxn];int n,ans[maxn];
8a63 /*注意需要按l将节点基数排序来拓扑更新parent树*/
3e3e struct Suffix_Automaton{
427e     //basic
0037     int nxt[maxn*2][26],fa[maxn*2],l[maxn*2];
0db0     int last,cnt;
427e     //extension
f6ac     int cntA[maxn*2],A[maxn*2];/*辅助拓扑更新*/
b0fc     int num[maxn*2];/*每个节点代表的所有串的出现次数*/
dd0f #ifdef RIGHT
0641     vector<int> E[maxn*2];
```

```
6561    int dfsl[maxn*2],dfsr[maxn*2],dfn;
4296    int pos[maxn*2];
efe5    int end_pos[maxn*2];//1基
1937 #endif
c75a    Suffix_Automaton(){ clear(); }
1126    void clear(){
651a        last =cnt=1;
63e2        fa[1]=l[1]=0;
9b85        memset(nxt[1],0,sizeof nxt[1]);
95cf    }
e798    void init(char *s){
f205        while (*s){
d3f9            add(*s-'a');s++;
95cf        }
95cf    }
681b    void add(int c){
a4cf        int p = last;
4428        int np = ++cnt;
8b9f        memset(nxt[cnt],0,sizeof nxt[cnt]);
97c0        l[np] = l[p]+1;last = np;
b7f5        while (p&&!nxt[p][c])nxt[p][c] = np,p = fa[p];
fdc4        if (!p)fa[np]=1;
037f        else{
5740            int q = nxt[p][c];
d84d            if (l[q]==l[p]+1)fa[np] =q;
037f            else{
2401                int nq = ++ cnt;
bc67                l[nq] = l[p]+1;
da26                memcpy(nxt[nq],nxt[q],sizeof (nxt[q]));
66a6                fa[nq] =fa[q];fa[np] = fa[q] =nq;
5dc1                while (nxt[p][c]==q)nxt[p][c] =nq,p = fa[p];
95cf            }
95cf        }
95cf    }
2114    void build(){
4006        memset(cntA,0,sizeof cntA);
7b40        memset(num,0,sizeof num);
1a84        for (int i=1;i<=cnt;i++)cntA[l[i]]++;
856c        for (int i=1;i<=cnt;i++)cntA[i]+=cntA[i-1];
ebb3        for (int i=cnt;i>=1;i--)A[cntA[l[i]]--] =i;
f42d        /*更行主串节点*/
3c9b        int temp=1;
1294        for (int i=0;i<n;i++){
3bd2            num[temp = nxt[temp][s[i]-'a'] ]=1;
95cf        }
e1a0        /*拓扑更新*/
5258        for (int i=cnt;i>=1;i--){
427e            //basic
b7fa            int x = A[i];
32d6            num[fa[x]]+=num[x];
427e            //special
f982            ans[l[x]] = max(ans[l[x]],num[x]);
95cf        }
427e        //special
66f2        for (int i=l[last];i>1;i--){
88a3            ans[i-1] = max(ans[i-1],ans[i]);
95cf        }
95cf    }
427e
dd0f #ifdef RIGHT
a1e1    int get_right_between(int u,int l,int r){
64ba        return tree.query(tree.root[dfsl[u] - 1],tree.root[dfsr[u]],1,::n,l,r);
95cf    }
d714    void dfs(int u){
2b56        dfsl[u] = ++ dfn;
98d9        pos[dfn] = u;
2c0f        for (int v : E[u]){
5f3c            dfs(v);
95cf        }
64a8        dfsr[u] = dfn;
95cf    }
0350    void extract_right(){
3c9b        int temp = 1;
1294        for (int i=0;i<n;i++){
ac16            temp = nxt[temp][s[i] - 'a'];
6940            end_pos[temp] = i+1;
95cf        }
f6b7        for (int i=2;i<=cnt;i++){
5e80            E[fa[i]].push_back(i);
95cf        }
0426        dfn = 0;
dcdd        dfs(1);
5087        tree.root[0] = tree.buildT0(1,n);
7b35        for (int i=1;i<=cnt;i++){
cda5            int u = pos[i];
1c34            if (end_pos[u]){
9965                int idx = end_pos[u];
b360                tree.root[i] = tree.update(tree.root[i-1],1,n,idx,1);
```

```
8e2e             }else{
d757                 tree.root[i] = tree.root[i-1];
95cf             }
95cf         }
95cf     }
1937 #endif
56dd     void debug(){
5258         for (int i=cnt;i>=1;i--){
01ab             printf("num[%d]=%d l[%d]=%d fa[%d]=%d\n",i,num[i],i,l[i],i,fa[i]);
95cf         }
95cf     }
5eed }sam;
3117 int main(){
587c     scanf("%s",s);
aaa0     /* calc n must before sam.init()*/
5264     n = strlen(s);
84b5     sam.init(s);
bb59     sam.build();
6dbf     for (int i=1;i<=n;i++){
6240         printf("%d\n",ans[i]);
95cf     }
7021     return 0;
95cf }
```

2.3 Generlized SAM

```
427e // Created by calabash_boy on 19-4-5.
427e //wf2019 first of her name
427e //build sam using trie
302f #include<bits/stdc++.h>
421c using namespace std;
94a1 const int maxn = 1e6+100;
4085 typedef long long ll;
3e3e struct Suffix_Automaton{
0037     int nxt[maxn*2][26],fa[maxn*2],l[maxn*2];
0db0     int last,cnt;
0641     vector<int> E[maxn*2];
61cb     int Num[maxn*2];
c75a     Suffix_Automaton(){ clear(); }
1126     void clear(){
651a         last =cnt=1;
63e2         fa[1]=l[1]=0;
9b85         memset(nxt[1],0,sizeof nxt[1]);
95cf     }
6cab     int add(int pre,int c,int num){
2d24         last = pre;
a4cf         int p = last;
4428         int np = ++cnt;
b844         Num[np] = num;
8b9f         memset(nxt[cnt],0,sizeof nxt[cnt]);
97c0         l[np] = l[p]+1;last = np;
b7f5         while (p&&!nxt[p][c])nxt[p][c] = np,p = fa[p];
fdc4         if (!p)fa[np]=1;
037f         else{
5740             int q = nxt[p][c];
d84d             if (l[q]==l[p]+1)fa[np] =q;
037f             else{
2401                 int nq = ++ cnt;
bc67                 l[nq] = l[p]+1;
da26                 memcpy(nxt[nq],nxt[q],sizeof (nxt[q]));
66a6                 fa[nq] =fa[q];fa[np] = fa[q] =nq;
5dc1                 while (nxt[p][c]==q)nxt[p][c] =nq,p = fa[p];
95cf             }
95cf         }
597e         return np;
95cf     }
b432     int dfsl[maxn*2],dfsr[maxn*2];
b4c2     int dfn = 0;
45bd     ll sum[maxn*2];
d714     void dfs(int u){
2b56         dfsl[u] = ++dfn;
445a         sum[dfn] = Num[u];
2c0f         for (int v : E[u]){
5f3c             dfs(v);
95cf         }
64a8         dfsr[u] = dfn;
95cf     }
2114     void build(){
f6b7         for (int i=2;i<=cnt;i++){
5e80             E[fa[i]].push_back(i);
95cf         }
dcdd         dfs(1);
7b35         for (int i=1;i<=cnt;i++){
036a             sum[i] += sum[i-1];
95cf         }
95cf     }
```

```
    void query(char * s){
        int temp = 1;
        while (*s){
            int ch = *s - 'A';
            if (!nxt[temp][ch]){
                printf("0\n");
                return;
            }
            temp = nxt[temp][ch];
            s++;
        }
        ll ans = sum[dfsr[temp]] - sum[dfsl[temp] - 1];
        printf("%lld\n",ans);
    }
}sam;
struct Trie{
    int Root = 1;
    int cnt = 2;
    int nxt[maxn][26];
    int num[maxn];
    int sam_pos[maxn];
    int add(int p,int ch){
        if (!nxt[p][ch]){
            nxt[p][ch] = cnt++;
        }
        int now = nxt[p][ch];
        num[now] ++;
        return now;
    }
    void bfs(){
        queue<int> Q;
        Q.push(1);
        sam_pos[1] = 1;
        while (!Q.empty()){
            int head = Q.front();
            Q.pop();
            for (int i=0;i<26;i++){
                if (!nxt[head][i])continue;
                int now = nxt[head][i];
                sam_pos[now] = sam.add(sam_pos[head],i,num[now]);
                Q.push(now);
            }
        }
    }
}trie;
int trie_pos[maxn];
int main(){
    int n,k;
    scanf("%d%d",&n,&k);
    trie_pos[0] = 1;
    for (int i=1;i<=n;i++){
        static char s[5];
        int p;
        scanf("%s%d",s,&p);
        int ch = s[0] - 'A';
        trie_pos[i] = trie.add(trie_pos[p],ch);
    }
    trie.bfs();
    sam.build();
    for (int i=0;i<k;i++){
        static char t[maxn];
        scanf("%s",t);
        int N = strlen(t);
        reverse(t,t+N);
        sam.query(t);
    }
    return 0;
}
```

2.4 PAM

```
// Created by calabash_boy on 18-6-4.
// BZOJ 3676
// calc max(len(t)*cnt(t)) t为s回文子串，cnt(t)=t出现次数
#include<bits/stdc++.h>
using namespace std;
const int maxn = 3e5+100;
struct Palindromic_AutoMaton{
    //basic
    int s[maxn],now;
    int nxt[maxn][26],fail[maxn],l[maxn],last,tot;
    // extension
    int num[maxn];/*节点代表的所有回文串出现次数*/
    void clear(){
        //1节点：奇数长度root 0节点：偶数长度root
        s[0]=l[1]=-1;
```

```
b6d0        fail[0] = tot = now =1;
f40b        last = l[0]=0;
21a1        memset(nxt[0],0,sizeof nxt[0]);
9b85        memset(nxt[1],0,sizeof nxt[1]);
95cf    }
61ff    Palindromic_AutoMaton(){clear();}
ca1c    int newnode(int ll){
71cf        tot++;
87f4        memset(nxt[tot],0,sizeof nxt[tot]);
dd2b        fail[tot]=num[tot]=0;
1621        l[tot]=ll;
91fb        return tot;
95cf    }
4284    int get_fail(int x){
8ef1        while (s[now-l[x]-2]!=s[now-1])x = fail[x];
d074        return x;
95cf    }
a791    void add(int ch){
3622        s[now++] = ch;
051b        int cur = get_fail(last);
a980        if(!nxt[cur][ch]){
80d2            int tt = newnode(l[cur]+2);
2f33            fail[tt] = nxt[get_fail(fail[cur])][ch];
01cb            nxt[cur][ch] = tt;
95cf        }
c2d8        last = nxt[cur][ch];num[last]++;
95cf    }
2114    void build(){
427e        //fail[i]<i，拓扑更新可以单调扫描。
0f06        for (int i=tot;i>=2;i--){
925b            num[fail[i]]+=num[i];
95cf        }
6b35        num[0]=num[1]=0;
95cf    }
2e3f    void init(char* ss){
36c9        while (*ss){
884f            add(*ss-'a');ss++;
95cf        }
95cf    }
d155    void init(string str){
10ad        for (int i=0;i<str.size();i++){
e6ef            add(str[i]-'a');
95cf        }
95cf    }
```

```
7b0e    long long query();
de71}pam;
26a1long long Palindromic_AutoMaton::query(){
8955    long long ret =1;
84e9    for (int i=2;i<=tot;i++){
e902        ret = max(ret,1LL*l[i]*num[i]);
95cf    }
ee0f    return ret;
95cf}
15dfchar s[maxn];
3117int main(){
587c    scanf("%s",s);
6780    pam.init(s);
bcac    pam.build();
baad    printf("%lld\n",pam.query());
7021    return 0;
95cf}
```

3 Algorithm

3.1 Geometry

```
302f#include <bits/stdc++.h>
421cusing namespace std;
ce18const int maxn = 10000 + 50;
320etemplate<class type>
9704struct point{
ce03    type x,y;
5cb2    point(){};
f40a    point(type x_,type y_):x(x_),y(y_){}
f510    point operator +(const point &p)const {return point(x + p.x,y + p.y);}
3ecb    point operator -(const point &p)const {return point(x - p.x,y - p.y);}
427e    //a related to b
427e    //clockwise : positive
427e    //anti-clockwise : negative
427e    //share a line : zero
dce3    type cross(const point &p)const {return x * p.y - y * p.x;}
a809    type dot(const point &p)const {return x * p.x + y * p.y;}
2f3a    type cross(const point &a,const point &b)const {return (a - *this).cross(b -
          *this);}
7f6b    type dot(const point &a,const point &b)const {return (a - *this).dot(b - *
          this);}
```

```
d92f     type sqrLen()const{return this->dot(*this);}
5bed     type sqrDis(const point &p)const {return (p - *this).sqrLen();}
329b };
d7b8 typedef point<long long> pt;
9d10 namespace Geometry{
fd78     const double PI = acos(-1.0);
427e     //res[0]: left most and bottom most
427e     //anti-clockwise
427e     //no three points share one line
427e     //WARN: this function modifies points
2325     vector<pt> Convex_Hull(vector<pt> &points){
8fa3         vector<pt> res(0);
0ca4         assert(points.size() >= 3);
bf80         int idx = 0;
6281         for (int i=1;i<points.size();i++){
28dc             pt temp = points[i];
a34c             pt now = points[idx];
4897             if (temp.x < now.x || temp.x == now.x && temp.y < now.y)idx = i;
95cf         }
8d08         swap(points[idx],points[0]);
9837         sort(points.begin()+1,points.end(),[&](pt x,pt y){
89c2             double cro = points[0].cross(x,y);
69ef             if (cro != 0)return cro > 0;
180e             return points[0].sqrDis(x) < points[0].sqrDis(y);
b251         });
7271         res.push_back(points[0]);
c57e         res.push_back(points[1]);
8316         for (int i=2;i<points.size();i++){
b7b9             pt now = points[i];
b94e             while (res.size() >= 2){
df0d                 double cro = res[res.size()-2].cross(now,res.back());
f72d                 auto p = res[res.size()-2];
e810                 auto pp = res.back();
63f2                 if (cro >= 0)res.pop_back();
caf8                 else break;
95cf             }
49f1             res.push_back(now);
95cf         }
244d         return res;
95cf     }
427e     //calc the Minkowski Sum of two Convex Hull
d0a9     vector<pt> Minkowski(const vector<pt> &ch1,const vector<pt> &ch2){
ef50         assert(ch1.size() >= 3);
ff7e         assert(ch2.size() >= 3);
7c15         stack<pt> vec1;
ee2b         stack<pt> vec2;
7245         for (int i = ch1.size() - 1;i >=0;i--){
a9f5             vec1.push(ch1[(i+1)%ch1.size()] - ch1[i]);
95cf         }
3cde         for (int i = ch2.size() - 1;i >= 0;i--){
6f4f             vec2.push(ch2[(i+1)%ch2.size()] - ch2[i]);
95cf         }
8fa3         vector<pt> res(0);
2219         res.push_back(ch1.front() + ch2.front());
186a         while (!vec1.empty() && !vec2.empty()){
b518             auto v1 = vec1.top();
f296             auto v2 = vec2.top();
dca9             long long cro = v1.cross(v2);
6b8d             if (cro > 0){
0c49                 res.push_back(res.back() + v1);
cb19                 vec1.pop();
8e2e             }else{
0ea2                 res.push_back(res.back() + v2);
fe8d                 vec2.pop();
95cf             }
95cf         }
6ca4         while (!vec1.empty())res.push_back(res.back() + vec1.top()),vec1.pop();
b356         while (!vec2.empty())res.push_back(res.back() + vec2.top()),vec2.pop();
1f73         return Convex_Hull(res);
95cf     }
427e     //logn
427e     //wether point in or on convex hull
a023     bool within(pt p,const vector<pt> &ch){
0c3b         assert(ch.size() >= 3);
5221         auto base = ch.front();
d6e7         if (base.cross(p,ch[1]) > 0 || base.cross(p,ch.back()) < 0)return false;
684c         if (base.cross(p,ch[1]) == 0 && (p - base).sqrLen() <= (ch[1] - base).
               sqrLen())return true;
265b         auto cmp = [&](const pt x,const pt y){
d8cd             long long cro = base.cross(x,y);
61b4             return cro>0;
329b         };
d4ae         int i = lower_bound(ch.begin(),ch.end(),p,cmp) - ch.begin() - 1;
8132         int j = i+1;
635b         assert(j < ch.size());
c740         return ch[i].cross(ch[j],p) >= 0;
95cf     }
329b };
```

3.2 Max Flow

```
427e // Created by calabash_boy on 18-9-14.
302f #include<bits/stdc++.h>
421c using namespace std;
4085 typedef long long ll;
32d7 const int maxn = 11000;
3378 const int maxm = 110000;
08a4 const int INF = 0x3f3f3f3f;
5650 struct Max_Flow{
f1b1     int first[maxn],nxt[maxm*2],des[maxm*2],c[maxm*2],tot;
4e95     int dep[maxn];int ss,tt;
b376     Max_Flow(){ clear(); }
1126     void clear(){
4e61         memset(first,-1,sizeof first);tot =-1;
95cf     }
4a69     inline void addEdge(int u,int v,int w){
71cf         tot++;
73e4         des[tot] = v;c[tot] =w;
6570         nxt[tot] = first[u];first[u] = tot;
95cf     }
1836     bool bfs(){
d568         memset(dep,-1,sizeof dep);
0881         dep[ss] =0;
fc6b         queue<int> Q;Q.push(ss);
11e5         while (!Q.empty()){
d7b1             int q = Q.front();Q.pop();
9c72             for (int t = first[q];t!=-1;t= nxt[t]){
b7bb                 int v = des[t],cx = c[t];
c804                 if (dep[v]==-1&&cx){
31e8                     dep[v] = dep[q]+1;
78e5                     Q.push(v);
95cf                 }
95cf             }
95cf         }
45fe         return dep[tt]!=-1;
95cf     }
c29e     int dfs(int node,int now){
0031         if (node==tt)return now;
5839         int res =0;
1e7e         for (int t = first[node];t!=-1&&res<now;t=nxt[t]){
b7bb             int v = des[t],cx = c[t];
da1a             if (dep[v]==dep[node]+1&&cx){
223c                 int x = min(cx,now-res);
6c2e                 x = dfs(v,x);
29d4                 res+=x;c[t]-=x;c[t^1]+=x;
95cf             }
95cf         }
7399         if (!res) dep[node] = -2;
244d         return res;
95cf     }
427e     // tuple<from,to,flow>
4649     void init(vector<tuple<int,int,int> > Edge){
1cbd         for (auto tp : Edge){
1de2             int u,v,w;tie(u,v,w) = tp;
16fe             addEdge(u,v,w);addEdge(v,u,0);
95cf         }
95cf     }
427e     // s->t max_flow
9783     ll max_flow(int s,int t){
8786         ss = s;tt = t;
692e         ll res =0,del =0;
75d3         while (bfs()){while (del = dfs(ss,INF)){res += del;}}
244d         return res;
95cf     }
8596 }net;
4dbf int n,m,s,t;
8f52 vector<tuple<int,int,int> > E;
3117 int main(){
5dae     scanf("%d%d%d%d",&n,&m,&s,&t);
356f     for (int i=0;i<m;i++){
3676         int u,v,w;
95a1         scanf("%d%d%d",&u,&v,&w);
be22         E.push_back(make_tuple(u,v,w));
95cf     }
08d9     net.init(E);
9560     printf("%lld\n",net.max_flow(s,t));
7021     return 0;
95cf }
```

3.3 Min Cost Max Flow(Min Cost Flow)

```
427e
```

```
427e // Created by calabash_boy on 19-10-5.
302f #include <bits/stdc++.h>
421c using namespace std;
6cca const int maxn = 3 * 250 + 100;
1517 const int maxm = 2 * 250 * 250 + 100;
b9bf const int inf = 10000;
08a4 const int INF = 0x3f3f3f3f;
c6cb struct MCMF{
5217     int ss,tt,dis[maxn],pre[maxn];
4b98     int first[maxn],from[maxm*2],des[maxm*2],nxt[maxm*2],cost[maxm*2],flow[maxm
           *2],tot;
e50d     bool in[maxn];
2826     MCMF(){
1d56         clear();
95cf     }
1126     void clear(){
ee65         tot =-1;
8eac         memset(first,-1,sizeof first);
95cf     }
427e     // <u,v,flow,cost>
d399     void init(vector<tuple<int,int,int,int> > E){
757c         for (auto edge : E){
4240             int u,v,f,c;
231d             tie(u,v,f,c) = edge;
b841             addEdge(u, v, f, c);
95cf         }
95cf     }
dbb4     void __addE(int x,int y,int f,int c){
71cf         tot++;
575f         from[tot] =x;des[tot] =y;
4b45         flow[tot] =f;cost[tot] =c;
6d84         nxt[tot] = first[x];first[x] = tot;
95cf     }
f1f8     inline void addEdge(int x,int y,int f,int c){
f355         __addE(x,y,f,c);__addE(y,x,0,-c);
95cf     }
3c52     bool spfa(){
f25d         memset(in,0,sizeof in);
a9d8         for (int i=0;i<maxn;i++)dis[i] = INF;
56b2         memset(pre,-1,sizeof pre);
9669         dis[ss] =0;in[ss] =1;
fc6b         queue<int> Q;Q.push(ss);
11e5         while (!Q.empty()){
3b29             int q = Q.front();
f56a             Q.pop();in[q] = 0;
9c72             for (int t = first[q];t!=-1;t = nxt[t]){
4993                 int v=des[t],len=cost[t],cx=flow[t];
50ae                 if (cx&&dis[v]>dis[q]+len){
e29b                     dis[v] = dis[q]+len;
0986                     pre[v] = t;
7476                     if (!in[v]){
d143                         Q.push(v);in[v] = 1;
95cf                     }
95cf                 }
95cf             }
95cf         }
95cf     }
427e     // min cost max flow
427e     //return pre[tt] != -1;
427e
427e     // min cost. flow needn't be max.
5287     return pre[tt]!=-1 && dis[tt] < 0;
95cf }
427e // <flow,cost>
ae82 pair<int,int> run(int s, int t){
8786     ss =s;tt=t;
eb96     int totflow =0,totcost =0,nowflow =0,nowcost =0;
22dc     while (spfa()){
2c90         nowcost =0;nowflow = INF;
d3ff         int now =pre[tt];
21b8         while (now!=-1){
f5f6             nowflow = min(nowflow,flow[now]);
61af             now = pre[from[now]];
95cf         }
83dd         now = pre[tt];
21b8         while (now!=-1){
1839             flow[now] -= nowflow;
fee0             flow[now^1] += nowflow;
96be             nowcost +=cost[now];
61af             now = pre[from[now]];
95cf         }
db07         nowcost*=nowflow;
9bc4         totflow +=nowflow;
0178         totcost +=nowcost;
95cf     }
9589     return make_pair(totflow,totcost);
95cf }
427e // special
0abd void output(int cost);
```

```
70ae }mcmf;
35b8 int n,m;
8960 int a[maxn];
e8ac int id[maxn];
5718 int argvalue[maxn];
a300 vector<string> ans;
c056 void copy(int argid,int val){
3970     stringstream stm;
2fb3     stm<<(char)('a' + argid - 1)<<"="<<val;
e0f6     ans.push_back(stm.str());
95cf }
2def void print(int argid){
3970     stringstream stm;
ab5f     stm<<"print("<<(char)('a' + argid - 1)<<")";
e0f6     ans.push_back(stm.str());
95cf }
5273 void MCMF::output(int cost){
610d     int argid = 0;
6dbf     for (int i=1;i<=n;i++){
3db1         int A = 2 * i-1;
fe76         int B = 2 * i;
3979         if (id[A] == 0){
dbc5             argid ++;
c40b             id[A] = argid;
a4ca             copy(argid, a[i]);
9257             print(argid);
79a3             argvalue[argid] = a[i];
8e2e         }else{
2c77             int temp_value = argvalue[id[A]];
080d             if (temp_value != a[i]){
16e6                 copy(id[A], a[i]);
8c83                 argvalue[id[A]] = a[i];
95cf             }
b391             print(id[A]);
95cf         }
2516         for (int t = first[B];t != -1;t = nxt[t]){
e8e0             int v = des[t];
2bc8             int f = flow[t];
c8f5             if (f|| v == A){
b333                 continue;
95cf             }
f914             if (v == 2 * n + 3) break;
037f             else{
8919                 id[v] = id[A];
95cf             }
95cf         }
95cf     }
6f76     cout<<ans.size()<<" "<<cost<<endl;
03de     for (auto str : ans){
cc6d         cout<<str<<endl;
95cf     }
95cf }
3117 int main(){
9af0     cin>>n>>m;
6dbf     for (int i=1;i<=n;i++){
879c         cin>>a[i];
95cf     }
efbd     vector<tuple<int,int,int,int> > E(0);
f385     int SS = 2 * n + 1;
dc84     int S = 2 * n + 2;
c8df     int T = 2 * n + 3;
6962     E.push_back(make_tuple(SS,S,m,0));
6dbf     for (int i=1;i<=n;i++){
3db1         int A = 2 * i - 1;
fe76         int B = 2 * i;
3531         E.push_back(make_tuple(A,B,1,-inf));
1cb5         E.push_back(make_tuple(S,A,1,__builtin_popcount(a[i])));
0673         E.push_back(make_tuple(B,T,1,0));
ed35         for (int j=i+1;j<=n;j++){
71ea             int AA = 2 * j - 1;
1e22             int BB = 2 * j;
084e             if (a[i] == a[j]){
6be3                 E.push_back(make_tuple(B,AA,1,0));
8e2e             }else{
782c                 E.push_back(make_tuple(B,AA,1,__builtin_popcount(a[j])));
95cf             }
95cf         }
95cf     }
2ec5     mcmf.init(E);
8f04     pair<int,int> ans = mcmf.run(SS, T);
427e     //cerr<<ans.first<<","<<ans.second<<endl;
61da     mcmf.output((ans.second% inf + inf) % inf);
7021     return 0;
95cf }
```

3.4 LCA

```
427e // Created by calabash_boy on 18-7-7.
302f #include<bits/stdc++.h>
421c using namespace std;
6f64 const int maxn = 5e5+100;
58a9 int first[maxn],des[maxn*2],nxt[maxn*2],tot;
53ee int n,m,s;
911d inline int addEdge(int x,int y){
4704     tot++;des[tot] = y;
465b     nxt[tot] = first[x];
86fa     first[x] = tot;
95cf }
22cd namespace Multiply_LCA{
ae22     int fa[maxn][20],dep[maxn];
2b4e     void dfs(int u,int father){
5620         fa[u][0] = father;
0b67         dep[u] = dep[father]+1;
1677         for (int i=1;i<20&&fa[u][i-1];i++){
9f44             fa[u][i] = fa[fa[u][i-1]][i-1];
95cf         }
3ddf         for (int t=first[u];t;t=nxt[t]){
e8e0             int v = des[t];
ca31             if (v==father)continue;
e2f7             dfs(v,u);
95cf         }
95cf     }
620b     int lca(int x,int y){
d22b         if (dep[x]<dep[y])swap(x,y);
1534         for (int i=19;i>=0;i--){
8ab5             if (dep[fa[x][i]]>=dep[y]){
ec54                 x = fa[x][i];
95cf             }
95cf         }
bb52         if (x==y)return x;
1534         for (int i=19;i>=0;i--){
c55c             if (fa[x][i]!=fa[y][i]){
ec54                 x = fa[x][i];
c413                 y = fa[y][i];
95cf             }
95cf         }
8fb3         return fa[y][0];
95cf     }
329b };
3117 int main(){
080c     scanf("%d%d%d",&n,&m,&s);
324a     for (int i=1;i<n;i++){
0f8b         int x,y;
a9b3         scanf("%d%d",&x,&y);
7487         addEdge(x,y);addEdge(y,x);
95cf     }
73b1     Multiply_LCA::dfs(s,0);
3f3a     while (m--){
bf62         int x,y;scanf("%d%d",&x,&y);
d93e         printf("%d\n",Multiply_LCA::lca(x,y));
95cf     }
7021     return 0;
95cf }
```

3.5 DSU On Tree

```
427e // Created by calabash_boy on 18-10-8.
427e // 1-rooted tree
427e // query vertex with height H in subtree of V
427e // whether the letter can form a palindrome
302f #include <bits/stdc++.h>
421c using namespace std;
4085 typedef long long ll;
3688 typedef pair<int,int> pii;
31ec #define rep(i,l,r) for (ll i = l,_ = r;i< _;i++)
5879 #define REP(i,l,r) for (ll i=l,_=r;i<=_;i++)
6f64 const int maxn = 5e5+100;
2ff9 int n,tot,first[maxn],des[maxn],nxt[maxn],m;
28d5 vector<pii> Q[maxn];
f96d int cnt[maxn][26],Cnt[maxn];
bbe3 int sz[maxn],dep[maxn],wson[maxn];
f0f2 bool ans[maxn],big[maxn];
15df char s[maxn];
453e inline void addEdge(int x,int y){
4704     tot++;des[tot] = y;
465b     nxt[tot] = first[x];
86fa     first[x] = tot;
95cf }
0d39 void get_sz(int node,int depth){
2b42     dep[node] = depth;sz[node] = 1;
e83e     for (int t = first[node];t;t=nxt[t]){
e8e0         int v = des[t];
```

```
a0d5         get_sz(v,depth+1);
47d5         sz[node] += sz[v];
03ee         if (sz[v] > sz[wson[node]])wson[node] = v;
95cf     }
95cf }
5efd void add(int node,int sign){
b01b     Cnt[dep[node]] -= cnt[dep[node]][s[node]-'a'];
d2e8     cnt[dep[node]][s[node]-'a'] ^=1;
937f     Cnt[dep[node]] += cnt[dep[node]][s[node]-'a'];
e83e     for (int t = first[node];t;t=nxt[t]){
e8e0         int v = des[t];
dcb7         if (big[v])continue;
ec6e         add(v,sign);
95cf     }
95cf }
5cc1 void dfs(int node,bool keep){
e83e     for (int t = first[node];t;t=nxt[t]){
e8e0         int v = des[t];
5279         if (v == wson[node])continue;
4bc1         dfs(v,0);
95cf     }
d010     if (wson[node]){
6048         big[wson[node]]=1;
11b7         dfs(wson[node],1);
95cf     }
7111     add(node,1);
3a0c     for (auto q:Q[node]){
1c95         ans[q.second] = Cnt[q.first] <=1;
95cf     }
918e     if (wson[node])big[wson[node]] = 0;
dc2a     if (!keep)add(node,-1);
95cf }
3117 int main(){
ac98     scanf("%d%d",&n,&m);
eeaf     REP(i,2,n){
4ec4         int p;
e75e         scanf("%d",&p);
be80         addEdge(p,i);
95cf     }
a275     scanf("%s",s+1);
a826     rep(i,0,m){
8213         int v,h;
fdd4         scanf("%d%d",&v,&h);
3e7f         Q[v].push_back({h,i});
95cf     }
ff05     get_sz(1,1);dfs(1,0);
8823     rep(i,0,m)printf("%s\n",ans[i]?"Yes":"No");
7021     return 0;
95cf }
```

4 Data Structure

4.1 01 Trie

```
427e // Created by calabash_boy on 18-7-7.
427e // max(XorSum(a_l^r))
302f #include<bits/stdc++.h>
421c using namespace std;
ed66 const int MAX = 1e6+100;
80de int bas[35],n,Cas;
92ad const int INF = 2147483645;
a281 struct Trie{
abd0     int nxt[MAX<<2][2],l[MAX<<2];
a945     int cnt,ansl,ansr,ansv;
5d53     void init(){
68de         cnt =ansv = 0;
16d8         memset(nxt[0],0,sizeof (nxt[0]));
aa76         memset(l,0x3f3f3f3f,sizeof(l));
95cf     }
b87c     int create(){
6fb3         cnt++;
3b79         memset(nxt[cnt],0,sizeof (nxt[cnt]));
6808         return cnt;
95cf     }
d5dd     void insert(int id,int x){
875c         int y = 0;
7ecf         for (int i=30;i>=0;i--){
0c9f             int t = x&bas[i];
2e46             t>>=i;
713f             if (!nxt[y][t])nxt[y][t] = create();
f056             y = nxt[y][t];
95cf         }
a4a7         l[y] = min(l[y],id);
95cf     }
1a97     void query(int id,int x){
537e         int y=0;  int res =0;
```

```
7ecf        for (int i=30;i>=0;i--){
0c9f            int t = x&bas[i];
2e46            t>>=i;
32ad            if (nxt[y][!t]){
63b9                y =nxt[y][!t];
1f38                res+=bas[i];
8e2e            }else{
f056                y = nxt[y][t];
95cf            }
95cf        }
181d        if (res==ansv){
a404            if (l[y]<ansl){
50d3                ansl = l[y];  ansr = id;
95cf            }
8135        }else if (res>ansv){
9429            ansv = res;
12f4            ansl = l[y];
37e9            ansr = id;
95cf        }
95cf    }
1cc7 }trie;
3117 int main(){
bf6d    bas[0] = 1;
1b53    for (int i1=1;i1<=30;i1++)bas[i1] = bas[i1-1]<<1;
3cb5    scanf("%d",&Cas);
3e2f    for (int i=1;i<=Cas;i++){
56d3        trie.init();  trie.insert(0,0);
cd91        scanf("%d",&n);
4d6a        int sum=0;
ede7        for (int j=1;j<=n;j++){
69e6            int ai;
3e9d            scanf("%d",&ai);  sum^=ai;
17a6            trie.query(j,sum);  trie.insert(j,sum);
95cf        }
7351        printf("Case␣#%d:\n%d␣%d\n", i, trie.ansl + 1, trie.ansr);
95cf    }
7021    return 0;
95cf }
```

4.2 Cartesian Tree

```
427e // Created by calabash_boy on 18-7-24.
```

```
427e //他的名字是笛卡尔树。
302f #include<bits/stdc++.h>
421c using namespace std;
1585 #define OPENSTACK
94a1 const int maxn = 1e6+100;
5d33 const int mod = 1e9+7;
5cad typedef long long LL;
a8dc int stk[maxn],top,sz[maxn];
8f18 int l[maxn],r[maxn],rt,n;
62bd pair<int,int> a[maxn];
2b49 LL inv[maxn],fac[maxn],inv_fac[maxn];
dbd8 bool vis[maxn];
ea2f /* l 左儿子 r 右儿子 rt根*/
2114 void build(){
3e5f    top=0;
4c1f    for (int i=1;i<=n;i++) l[i]=r[i]=vis[i] =0;
6dbf    for (int i=1;i<=n;i++){
8077        int k = top;
14fa        while (k&&a[i]<a[stk[k-1]])k--;
004e        if (k) r[stk[k-1]] = i;
90d1        if (k<top) l[i] = stk[k];
c046        stk[k++] =i;top = k;
95cf    }
791b    for (int i=1;i<=n;i++) vis[l[i]] = vis[r[i]] =1;
6dbf    for (int i=1;i<=n;i++){
794b        if (!vis[i]){
cf39            rt = i;
6173            break;
95cf        }
95cf    }
95cf }
a89a LL power(LL x,LL y){
0aee    LL res =1;
db1a    while (y){
349b        if (y&1)res = res*x%mod;
af39        y>>=1;
df96        x = x*x%mod;
95cf    }
244d    return res;
95cf }
0f81 inline LL C(int n,int m){
54dd    return fac[n]*inv_fac[m]%mod*inv_fac[n-m]%mod;
95cf }
f33f int dfs(int u){
```

```
    sz[u]=1;int ans =1;
    if (l[u])ans=1LL*ans*dfs(l[u])%mod;
    if (r[u])ans = 1LL*ans*dfs(r[u])%mod;
    sz[u]+=sz[l[u]]+sz[r[u]];
    return 1LL*ans*C(sz[u]-1,sz[l[u]])%mod;
}
void Main(){
    inv[1]=fac[1]=fac[0]=1;
    for (int i=2;i<maxn;i++)fac[i] = fac[i-1]*i%mod,inv[i] = inv[mod%i]*(mod-mod
      /i)%mod;
    inv_fac[maxn-1] = power(fac[maxn-1],mod-2);
    for (int i=maxn-2;i>=0;i--){
        inv_fac[i] = inv_fac[i+1]*(i+1)%mod;
    }
    int T;scanf("%d",&T);
    while (T--) {
        scanf("%d", &n);
        for (int i = 1; i <= n; i++) {
            int x;scanf("%d", &x);
            a[i] = {-x, i};
        }
        build();
        printf("%d\n", inv[2] * n % mod * power(fac[n], mod - 2) % mod * dfs(rt)
           % mod);
    }
}
int main(){
#ifdef OPENSTACK
    int size = 70 << 20; // 256MB
    char *p = (char*)malloc(size) + size;
#if (defined _WIN64) or (defined __unix)
    __asm__("movq␣%0,␣%%rsp\n" :: "r"(p));
#else
    __asm__("movl␣%0,␣%%esp\n" :: "r"(p));
#endif
#endif
    Main();
#ifdef OPENSTACK
    exit(0);
#else
    return 0;
#endif
}
```

4.3 Chairman Tree

```
// Created by calabash_boy on 18-7-7.
// query_kth_element
#include<bits/stdc++.h>
using namespace std;
const int maxn=1e5+100;
int a[maxn];int rk[maxn];int pos[maxn];
int root[maxn];int cnt,m,n,T;
struct Chairman_Tree{
    struct Node{int L,R,val;}tree[maxn*500];
    void init(){
        memset(root,0,sizeof root);
        cnt =0;
    }
    /* 建T0空树 */
    int buildT0(int l, int r){
        int k = cnt++;
        tree[k].val =0;
        if (l==r) return k;
        int mid = l+r >>1;
        tree[k].L = buildT0(l, mid);tree[k].R = buildT0(mid + 1, r);
        return k;
    }
    /* 上一个版本节点P, 【ppos】+=del 返回新版本节点*/
    int update (int P,int l,int r,int ppos,int del){
        int k = cnt++;
        tree[k].val = tree[P].val +del;
        if (l==r) return k;
        int mid = l+r >>1;
        if (ppos<=mid){
            tree[k].L = update(tree[P].L,l,mid,ppos,del);
            tree[k].R = tree[P].R;
        }else{
            tree[k].L = tree[P].L;
            tree[k].R = update(tree[P].R,mid+1,r,ppos,del);
        }
        return k;
    }
    int query_kth(int lt,int rt,int l,int r,int k){
        if (l==r) return a[rk[l]];
        int mid = l+r >>1;
        if (tree[tree[rt].L].val-tree[tree[lt].L].val>=k) return query_kth(tree[
          lt].L,tree[rt].L,l,mid,k);
```

```
38e4        else return query_kth(tree[lt].R,tree[rt].R,mid+1,r,k+tree[tree[lt].L].
              val-tree[tree[rt].L].val);
95cf    }
b0c1 }tree;
56b1 bool cmp(int x,int y){return a[x]<a[y];}
3117 int main() {
1fd9    scanf("%d", &T);
60ca    while (T--) {
ac98        scanf("%d%d",&n,&m);
6dbf        for (int i=1;i<=n;i++){
9a1c            scanf("%d",&a[i]);
f9d0            rk[i]=i;
95cf        }
a475        tree.init();
f0ca        sort(rk+1,rk+1+n,cmp);
8b31        for (int i1=1;i1<=n;i1++){
9b5e            pos[rk[i1]] =i1;
95cf        }
b6a2        root[0] = tree.buildT0(1, n);
8b31        for (int i1=1;i1<=n;i1++){
8294            root[i1] = tree.update(root[i1-1],1,n,pos[i1],1);
95cf        }
3f3a        while (m--){
d32c            int l,r,k;scanf("%d%d%d",&l,&r,&k);
26ab            printf("%d\n",tree.query_kth(root[l-1],root[r],1,n,k));
95cf        }
95cf    }
7021    return 0;
95cf }
```

4.4 KD Tree

```
427e // Created by calabash_boy on 18-10-6.
302f #include<bits/stdc++.h>
421c using namespace std;
5cad typedef long long LL;
eb45 const int maxn = 2e5+100;
b1ec const LL INF = 0x3f3f3f3f3f3f3f3fLL;
4d9b int m,n;
fc74 const int demension = 2;
4825 struct Hotel{
b199    int pos[demension],id,c;
}hotel[maxn],kdtree[maxn];                                                    4922
double var[demension];                                                        2ece
int split [maxn];int cmpDem;                                                  8003
bool cmp(const Hotel &a,const Hotel &b){                                      5cdc
    return a.pos[cmpDem]<b.pos[cmpDem];                                       b5cd
}                                                                             95cf
void build (int l,int r){                                                     d5af
    if (l>=r)return;                                                          2625
    int mid = l+r >>1;                                                        b8b7
    for (int i=0;i<demension;i++){                                            8037
        double ave =0;                                                        4655
        for (int j=l;j<=r;j++){                                               a0d3
            ave+=hotel[j].pos[i];                                             70b6
        }                                                                     95cf
        ave/=(r-l+1);var[i] =0;                                               b1eb
        for (int j=l;j<=r;j++){                                               a0d3
            var[i]+=pow(hotel[j].pos[i]-ave,2);                               27fe
        }                                                                     95cf
        var[i]/=(r-l+1);                                                      6e08
    }                                                                         95cf
    split[mid] =-1;double maxVar=-1;                                          3909
    for (int i=0;i<demension;i++){                                            8037
        if (var[i]>maxVar){                                                   d704
            maxVar = var[i];                                                  3bdc
            split[mid] =i;                                                    9c04
        }                                                                     95cf
    }                                                                         95cf
    cmpDem = split[mid];                                                      82fa
    nth_element(hotel+l,hotel+mid,hotel+r+1,cmp);                             d815
    build (l,mid-1);build (mid+1,r);                                          7bac
}                                                                             95cf
int ansIndex;                                                                 b10a
LL ansDis;                                                                    5721
void query(int l,int r,const Hotel& x){                                       c274
    if (l>r)return ;                                                          8b8a
    int mid = l+r >>1;LL dis =0;                                              c410
    for (int i=0;i<demension;i++){                                            8037
        dis +=1LL*(x.pos[i]-hotel[mid].pos[i])*(x.pos[i]-hotel[mid].pos[i]);  3cc8
    }                                                                         95cf
    if (hotel[mid].c<=x.c){                                                   9fff
        if (ansDis == dis && hotel[mid].id<hotel[ansIndex].id){               6bed
            ansIndex = mid;                                                   f191
        }else if (dis<ansDis){                                                f598
            ansDis = dis;                                                     de61
```

```
f191            ansIndex = mid;
95cf        }
95cf    }
fcd6    int d = split[mid];
78bf    LL radius = 1LL*(x.pos[d]-hotel[mid].pos[d])*(x.pos[d]-hotel[mid].pos[d]);
7ce7    if (x.pos[d]<hotel[mid].pos[d]){
8301        query(l,mid-1,x);
f036        if (ansDis>radius){query(mid+1,r,x);}
8e2e    }else{
32f9        query(mid+1,r,x);
6b1f        if (ansDis>radius){query(l,mid-1,x);}
95cf    }
95cf }
9523 int T;
0e91 void input(){
ac98    scanf("%d%d",&n,&m);
1294    for (int i=0;i<n;i++){
35bd        scanf("%d%d%d",&hotel[i].pos[0],&hotel[i].pos[1],&hotel[i].c);
cafc        hotel[i].id=i;
95cf    }
d489    build (0,n-1);
95cf }
9627 void solve(){
1a18    Hotel x;
e052    for (int i=1;i<=m;i++){
7fc9        scanf("%d%d%d",&x.pos[0],&x.pos[1],&x.c);
94af        ansDis = INF;ansIndex =n+1;
9760        query(0,n-1,x);
b64e        printf("%d␣%d␣%d\n",hotel[ansIndex].pos[0],hotel[ansIndex].pos[1],hotel[
              ansIndex].c);
95cf    }
95cf }
3117 int main(){
1fd9    scanf("%d",&T);
60ca    while (T--){
2a5c        input();
ccd1        solve();
95cf    }
7021    return 0;
95cf }
```

4.5 Segment Tree

```
427e // Created by calabash_boy on 18-9-14.
427e // interval modify & interval query
1915 #include<stdio.h>
421c using namespace std;
52c1 const int maxn = 1e5+100;
5cad typedef long long LL;
8960 int a[maxn];
b92c struct Seg_Tree{
b3d3    LL val[maxn*4];LL lazy[maxn*4];
77a4    inline void Up(int x){val[x] = val[x<<1]+val[x<<1|1];}
f043    inline void Down(int x,int l,int mid,int r){
7b86        if (lazy[x]){
777c            val[x<<1] += 1LL*lazy[x]*(mid-l+1);
664d            val[x<<1|1] += 1LL*lazy[x]*(r-mid);
5c48            lazy[x<<1]+= lazy[x];
dd43            lazy[x<<1|1] += lazy[x];
6cac            lazy[x] =0;
95cf        }
95cf    }
b1fe    void build (int x,int l,int r){
6cac        lazy[x] =0;
bcdf        if (l==r){val[x] = a[l];return ;}
b8b7        int mid = l+r >>1;
b3e3        build (x<<1,l,mid);build (x<<1|1,mid+1,r);
8eb6        Up(x);
95cf    }
f3fe    void add(int x,int l,int r,int L,int R,int del){
2fdc        if (l>R||r<L)return;
4d29        if (L<=l&&r<=R){
6171            val[x]+=1LL*del*(r-l+1);
1eeb            lazy[x]+=del;
4f2d            return;
95cf        }
b8b7        int mid = l+r >>1;
4dc2        Down(x,l,mid,r);
5468        add(x<<1,l,mid,L,R,del);add(x<<1|1,mid+1,r,L,R,del);
8eb6        Up(x);
95cf    }
073d    LL query_Sum(int x,int l,int r,int L,int R){
0872        if (l>R||r<L)return 0;
26cd        if (L<=l&&r<=R)return val[x];
b8b7        int mid = l+r >>1;
4dc2        Down(x,l,mid,r);
```

```
1fb2         return query_Sum(x<<1,l,mid,L,R)+query_Sum(x<<1|1,mid+1,r,L,R);
95cf     }
b0c1 }tree;
3d22 char opt[5];int m,n;
3117 int main(){
ac98     scanf("%d%d",&n,&m);
6dbf     for (int i=1;i<=n;i++){
60cb         scanf("%d",a+i);
95cf     }
e703     tree.build(1,1,n);
3f3a     while (m--){
42ba         int l,r,v;
e158         scanf("%s%d%d",opt,&l,&r);
0d1b         if (opt[0]=='Q'){
b8ef             printf("%I64d\n",tree.query_Sum(1,1,n,l,r));
ff96         }else if (opt[0]=='C'){
a9ba             scanf("%d",&v);
b937             tree.add(1,1,n,l,r,v);
95cf         }
95cf     }
7021     return 0;
95cf }
```

4.6 AFL(Cactus)

```
427e // Created by calabash_boy on 18-9-14.
427e // circle-square-tree Maximum independent set
302f #include<bits/stdc++.h>
421c using namespace std;
52c1 const int maxn = 1e5+100;
9010 vector<int> E1[maxn],ET[maxn];
c940 int m,n,N,fa[maxn],dp[maxn][2];
d746 int len[maxn],dfn[maxn],dfs_clock;
e6da bool inCircle[maxn];
4ab4 int dp2[maxn][2];
e227 inline void addEdge1(int x,int y){
f4a7     E1[x].push_back(y);
95cf }
2a27 inline void addEdgeT(int x,int y){
de38     ET[x].push_back(y);
95cf }
0e91 void input(){
64f1     cin>>n>>m;N =n;
356f     for (int i=0;i<m;i++){
97c3         int u,v;cin>>u>>v;
2775         addEdge1(u,v);addEdge1(v,u);
95cf     }
95cf }
74b1 void tarjan(int u){
f5c7     dfn[u] = ++dfs_clock;
1958     for (int i=0;i<E1[u].size();i++){
1654         int v = E1[u][i];
8e32         if (v==fa[u])continue;
3c64         if (!dfn[v]){
da94             fa[v] = u;tarjan(v);
e245         }else if (dfn[v]<dfn[u]){
c93c             n++;
478b             len[n] = dfn[u]-dfn[v]+1;
0f08             fa[n] = v;
92b2             addEdgeT(v,n);
8845             int temp = u;
a7eb             while (temp!=v){
3d33                 inCircle[temp] = true;
96c4                 addEdgeT(n,temp);
6dbe                 temp = fa[temp];
95cf             }
95cf         }
95cf     }
aeb9     if (!inCircle[u]){
6225         addEdgeT(fa[u],u);
95cf     }
e88e     dfs_clock--;
95cf }
662c void work(int x){
7330     int sz = ET[x].size();
03f3     if (sz==2){
bc63         int son1 = ET[x][0];
e1e3         int son2 = ET[x][1];
ff53         dp[x][0] = dp[son1][0]+dp[son2][0];
95d6         dp[x][1] = max(dp[son1][0]+dp[son2][0],max(dp[son1][0]+dp[son2][1],dp[
               son1][1]+dp[son2][0]));
4f2d         return;
95cf     }
3bde     dp2[0][0] =dp[ET[x][0]][0];dp2[0][1]=0;
e123     for (int i=1;i<sz;i++){
1022         dp2[i][0] = max(dp2[i-1][0],dp2[i-1][1])+dp[ET[x][i]][0];
```

```
6ecd         dp2[i][1] = dp2[i-1][0]+dp[ET[x][i]][1];
95cf     }
b6ba     dp[x][0] = dp2[sz-1][0];
cfc2     dp[x][1] = dp2[sz-1][0];
3347     dp2[sz][0]=dp2[sz][1]=0;
ca21     for (int i=sz-1;i>=0;i--){
858a         dp2[i][0] = max(dp2[i+1][0],dp2[i+1][1])+dp[ET[x][i]][0];
6f8c         dp2[i][1] = dp2[i+1][0]+dp[ET[x][i]][1];
95cf     }
5e56     dp[x][1] = max(dp[x][1],max(dp2[0][0],dp2[0][1]));
95cf }
d714 void dfs(int u){
0799     dp[u][0]=0;dp[u][1]=1;
16e7     if (u>N)dp[u][0]=0;
5ee5     for (int i=0;i<ET[u].size();i++){
f37f         int v = ET[u][i];
5f3c         dfs(v);
2900         if (u<=N){
edd9             dp[u][0]+=max(dp[v][1],dp[v][0]);
2a1b             dp[u][1]+=dp[v][0];
95cf         }
95cf     }
3200     if (u>N)work(u);
95cf }
3117 int main(){
2a5c     input();
951d     tarjan(1);
dcdd     dfs(1);
09a1     cout<<max(dp[1][0],dp[1][1])<<endl;
7021     return 0;
95cf }
```

4.7 Segment Tree(Dynamic Memory).cpp

```
427e // Created by calabash_boy on 18-10-1.
427e // CF 1046A
427e // give n tuple(x,r,p) and k<=20 , calc unordered pair(i,j)
427e // xi - ri <= xj <= xi + ri
427e // xj - rj <= xi <= xj + rj
427e // |pi - pj| <=k
302f #include <bits/stdc++.h>
421c using namespace std;
```

```
52c1 const int maxn = 1e5+100;
4085 typedef long long ll;
1c06 struct Node{ int L,R,val; }tree[maxn*200];
9f58 int cnt;
9c29 struct Segment_Tree{
e7b0     int root = 0;
ee91     int newnode(){
06cb         ++cnt;
6598         tree[cnt].val = tree[cnt].L = tree[cnt].R = 0;
6808         return cnt;
95cf     }
1483     Segment_Tree(){ root = newnode(); }
74ce     void add(int x,int l,int r,int Pos,int delta){
df5d         tree[x].val += delta;
0eec         if (l == r)return;
b8b7         int mid = l+r >>1;
5411         if (Pos <= mid){
88c7             if (tree[x].L == 0){
9efd                 tree[x].L = newnode();
95cf             }
55fc             add(tree[x].L,l,mid,Pos,delta);
8e2e         }else{
e74e             if (tree[x].R == 0){
ffbb                 tree[x].R = newnode();
95cf             }
492e             add(tree[x].R,mid+1,r,Pos,delta);
95cf         }
95cf     }
30b1     int query(int x,int l,int r,int L,int R){
52df         if (!x)return 0;
b8e7         if (l>R || L>r)return 0;
c450         if (L <= l && r <= R)return tree[x].val;
b8b7         int mid = l+r >>1;
b018         return query(tree[x].L,l,mid,L,R) + query(tree[x].R,mid+1,r,L,R);
95cf     }
329b };
9c0b map<int,Segment_Tree> mp;
9a6f map<int,int> id;
d7af int N;
3117 int main(){
232a     int n,k;
9927     scanf("%d%d",&n,&k);
ad91     vector<tuple<int,int,int> > a(n);
7739     vector<int> nums;
```

```
1294    for (int i=0;i<n;i++){
6a6b        int x,r,q;scanf("%d%d%d",&x,&r,&q);
82fb        a[i] = make_tuple(x,r,q);
3bee        nums.push_back(x);
ca6f        nums.push_back(x+r);
4730        nums.push_back(x-r);
95cf    }
19cd    sort(nums.begin(),nums.end());
e5bf    nums.erase(unique(nums.begin(),nums.end()),nums.end());
9e70    for (int i=0;i<nums.size();i++){
9b07        id[nums[i]] = i+1;
95cf    }
34ee    N = nums.size();
4c8a    sort(a.begin(),a.end(),[](const tuple<int,int,int> &a,const tuple<int,int,
          int>&b){
ddfb        return get<1>(a) > get<1>(b);
b251    });
19f3    ll ans =0;
1294    for (int i=0;i<n;i++){
2f4e        int x,r,q;tie(x,r,q) = a[i];
a8aa        int L = id[x-r],R = id[x+r];
af5f        for (int j=q-k;j<=q+k;j++){
7cd6            if (mp.find(j) == mp.end())continue;
8341            Segment_Tree & tree = mp[j];
e7d3            int root = tree.root;
768d            ans += tree.query(root,1,N,L,R);
95cf        }
e2c3        Segment_Tree & tree = mp[q];
e7d3        int root = tree.root;
9252        tree.add(root,1,N,id[x],1);
95cf    }
d592    cout<<ans<<endl;
7021    return 0;
95cf}
```

4.8 Rollback DSU

```
427e//加边删边二部图判定。
302f#include <bits/stdc++.h>
421cusing namespace std;
f374const int maxn = 1e5 + 20;
bd89struct UFS{
33ef    int fa[maxn];
590c    int sz[maxn];
6873    int len[maxn];
65fd    stack<pair<int*,int> > stk;
5d53    void init(){
e4ba        for (int i=1;i<maxn;i++){
974c            fa[i] = i;
fa1a            sz[i] = 1;
c008            len[i] = 0;
95cf        }
95cf    }
e034    UFS(){
07e2        init();
95cf    }
fee7    pair<int,int> find(int x){
7eb8        if (fa[x] == x)return make_pair(x,0);
037f        else{
2890            pair<int,int> ret = find(fa[x]);
22aa            ret.second ^= len[x];
ee0f            return ret;
95cf        }
95cf    }
427e    // 0 fail
427e    // 1 succ but not update
427e    // 2 succ and update
41b9    int merge(int x,int y){
7121        int fx,lenx;
5d92        int fy,leny;
9726        tie(fx,lenx) = find(x);
d13a        tie(fy,leny) = find(y);
e94b        if (fx == fy){
4350            return lenx ^ leny;
95cf        }
93ac        if (sz[fx] > sz[fy]){
65b4            swap(lenx,leny);
47d4            swap(x,y);
6c4f            swap(fx,fy);
95cf        }
dfaa        stk.push(make_pair(&sz[fy],sz[fy]));
863a        stk.push(make_pair(&fa[fx],fa[fx]));
a93a        fa[fx] = fy;
24e9        sz[fy] += sz[fx];
3c8a        if (lenx == leny){
5f4d            len[fx] = 1;
```

```
8e2e        }else{
7cc4            len[fx] = 0;
95cf        }
ca92        return 2;
95cf    }
831d    void rollback(){
5a7d        for (int i=0;i<2;i++){
503e            int * tar;
d26b            int val;
5b9a            tie(tar,val) = stk.top();
75b6            stk.pop();
9133            (*tar) = val;
95cf        }
427e
95cf    }
5795 }ufs;
58c6 const char* YES = "YES";
a0f7 const char* NO = "NO";
cd1e bool ans[maxn];
23cc struct SegmentTree{
90fc    vector<pair<int,int> > edges[maxn*4];
2161    void put(int x,int l,int r,int L,int R,pair<int,int> e){
d499        if (l > R || L > r)return;
4d29        if (L <= l && r <= R){
5bfc            edges[x].push_back(e);
4f2d            return;
95cf        }
b8b7        int mid = l + r >> 1;
8d76        put(x<<1,l,mid,L,R,e);
36cd        put(x<<1|1,mid+1,r,L,R,e);
95cf    }
8b28    void dfs(int x,int l,int r){
cd24        int succ = true;
8abb        int cnt = 0;
92f7        for (auto e : edges[x]){
0f8b            int x,y;
2bba            tie(x,y) = e;
6848            int ret = ufs.merge(x, y);
ecd5            succ &= ret!= 0;
7c6f            if (!succ){
9102                for (int i=0;i<cnt;i++)
5e31                    ufs.rollback();
4f2d                return;
95cf            }
feaf            cnt += ret == 2;
95cf        }
3a0d        if (l == r){
91cd            ans[l] = succ;
9102            for (int i=0;i<cnt;i++)
5e31                ufs.rollback();
4f2d            return;
95cf        }
b8b7        int mid = l + r >> 1;
7405        dfs(x<<1,l,mid);
b115        dfs(x<<1|1,mid+1,r);
9102        for (int i=0;i<cnt;i++)
5e31            ufs.rollback();
95cf    }
1d91    void debug(int x,int l,int r){
4bde        cerr<<x<<"␣:␣"<<"["<<l<<"␣,␣"<<r<<"␣]"<<endl;
92f7        for (auto e : edges[x]){
54f1            int u,v;
4c70            tie(u,v) = e;
40e5            cerr<<"<␣"<<u<<"␣,␣"<<v<<"␣>"<<endl;
95cf        }
0eec        if (l == r)return;
b8b7        int mid = l + r >> 1;
7dab        debug(x<<1,l,mid);
f599        debug(x<<1|1,mid+1,r);
95cf    }
f7fb }segtree;
ae0e map<pair<int,int>,vector<int> > mp;
3117 int main(){
1ed7    int n,q;
9c97    cin>>n>>q;
949d    for (int i=1;i<=q;i++){
54f1        int u,v;
a02c        cin>>u>>v;
fd0e        if (u > v)swap(u,v);
7c88        mp[make_pair(u,v)].push_back(i);
95cf    }
957e    for (auto pr : mp){
9660        vector<int> & ts = pr.second;
1e87        if (ts.size() & 1){
a1b6            ts.push_back(q+1);
95cf        }
a8d5        for (int i=0;i<ts.size();i+=2){
7ff9            int st = ts[i];
```

```
            int ed = ts[i+1] - 1;
            segtree.put(1, 1, q, st, ed, pr.first);
        }
    }
    //  segtree.debug(1,1,q);
    segtree.dfs(1, 1, q);
    for (int i=1;i<=q;i++){
        puts(ans[i]?YES:NO);
    }
    return 0;
}
```

5 Graph

5.1 Tarjan(BCC of Edge)

```
// Created by calabash_boy on 18-10-10.
#include<bits/stdc++.h>
using namespace std;
const int maxn = 1e5+100;
int first[maxn],nxt[maxn*2],from[maxn*2],des[maxn*2],isBrige[maxn*2],tot;
int dfn[maxn],low[maxn],dfs_clock;
int cnt_e[maxn],cnt_n[maxn];int bcc_cnt;
bool ok[maxn];vector <int> ans;int m,n;
inline void addEdge(int x,int y){
    tot++;
    des[tot] =y;from[tot] =x;
    nxt[tot] = first[x];first[x] = tot;
}
void input(){
    cin>>n>>m;
    for (int i=0;i<m;i++){
        int u,v;scanf("%d%d",&u,&v);
        addEdge(u,v);addEdge(v,u);
    }
}
void dfs(int u,int fa){
    dfn[u] = low[u] = ++dfs_clock;
    for (int t = first[u];t;t=nxt[t]){
        int v = des[t];if (v==fa)continue;
        if (!dfn[v]){
            dfs(v,u);
            low[u] = min(low[v],low[u]);
            if (dfn[u]<low[v]){
                isBrige[t] = true;
                if (t&1){isBrige[t+1] = true;}
                else{isBrige[t-1] = true;}
            }
        }else if (dfn[v]<dfn[u]){low[u] = min(low[u],dfn[v]);}
    }
}
void blood_fill(int x){
    dfn[x] = bcc_cnt;
    for (int t = first[x];t;t=nxt[t]){
        if (isBrige[t])continue;
        int v = des[t];
        if (!dfn[v]){blood_fill(v);}
    }
}
void check(){
    for (int i=1;i<=n;i++){cnt_n[dfn[i]]++;}
    for (int i=1;i<=tot;i++){
        if (isBrige[i]) continue;
        cnt_e[dfn[des[i]]]++;
    }
    for (int i=1;i<=bcc_cnt;i++){
        if (cnt_n[i]*2==cnt_e[i]){ok[i]=1;}
    }
}
void output(){
    for (int i=1;i<=tot;i+=2){
        if (isBrige[i])continue;
        if (ok[dfn[des[i]]])ans.push_back((i+1)/2);
    }
    sort(ans.begin(),ans.end());
    cout<<ans.size()<<endl;
    for (int i=0;i<ans.size();i++){printf("%d ",ans[i]);}
}
void solve(){
    for (int i=1;i<=n;i++){if (!dfn[i])dfs(i,-1);}
    memset(dfn,0,sizeof dfn);
    for (int i=1;i<=n;i++){
        if (!dfn[i]){
            bcc_cnt++;
            blood_fill(i);
        }
```

```
95cf     }
92ea     check();output();
95cf }
3117 int main(){
2a5c     input();
ccd1     solve();
7021     return 0;
95cf }
```

5.2 Tarjan(BCC of Point)

```
427e // Created by calabash_boy on 18-10-10.
302f #include<bits/stdc++.h>
421c using namespace std;
52c1 const int maxn = 1e5+100;
58a9 int first[maxn],des[maxn*2],nxt[maxn*2],tot;
09ab int bcc_cnt,cnt_n[maxn],cnt_e[maxn],bcc_no[maxn];
ff12 int dfn[maxn],low[maxn],dfs_clock;
8882 int st[maxn*2],top;bool ok[maxn];
5013 vector<int> ans;vector<int> temp;
4d9b int m,n;
453e inline void addEdge(int x,int y){
4704     tot++;des[tot] = y;
6d84     nxt[tot] = first[x];first[x] = tot;
95cf }
0e91 void input(){
9af0     cin>>n>>m;
356f     for (int i=0;i<m;i++){
17be         int u,v;scanf("%d%d",&u,&v);
ad4e         addEdge(u,v);addEdge(v,u);
95cf     }
95cf }
312b void dfs(int u,int fa){
d413     dfn[u] = low[u] = ++dfs_clock;
3ddf     for (int t = first[u];t;t=nxt[t]){
e8e0         int v = des[t];
b6ee         if (v==fa)continue;
3c64         if (!dfn[v]){
5248             st[top++] = t;dfs(v,u);
a19f             low[u] = min(low[u],low[v]);
9cb7             if (low[v]>=dfn[u]){
9d83                 bcc_cnt++;ok[bcc_cnt] = true;
1a7e                 temp.clear();
1026                 while (true){
87f2                     int tt = st[--top];
0648                     temp.push_back((tt+1)/2);
cf0f                     if (bcc_no[des[tt]]!=bcc_cnt){
aff7                         bcc_no[des[tt]] = bcc_cnt;
3e93                         cnt_n[bcc_cnt]++;
8e2e                     }else{
e551                         ok[bcc_cnt] = false;
95cf                     }
83bb                     cnt_e[bcc_cnt]++;
5047                     if (tt==t)break;
95cf                 }
b114                 if (ok[bcc_cnt]&&temp.size()>1){
af9b                     for (int i=0;i<temp.size();i++){
90d3                         ans.push_back(temp[i]);
95cf                     }
95cf                 }
95cf             }
e245         }else if (dfn[v]<dfn[u]){
be8d             st[top++] = t;
769a             low[u] = min(low[u],dfn[v]);
95cf         }
95cf     }
95cf }
9627 void solve(){
c2a0     for (int i=1;i<=n;i++){if (!dfn[i])dfs(i,-1);}
e139     sort(ans.begin(),ans.end());
c4d5     cout<<ans.size()<<endl;
263e     for (int i=0;i<ans.size();i++){printf("%d ",ans[i]);}
95cf }
3117 int main(){
2a5c     input();
ccd1     solve();
7021     return 0;
95cf }
```

5.3 Tarjan(SCC)

```
302f #include<bits/stdc++.h>
421c using namespace std;
52c1 const int maxn = 1e5+100;
```

```
04f1 int m,n,h;int t[maxn];
7560 int first[maxn*2],nxt[maxn*2],des[maxn*2],tot;
eaf3 int dfn[maxn],low[maxn],dft;bool d[maxn];
414b int flag[maxn],cnt[maxn],scc;stack<int> stk;
e50d bool in[maxn];
704e inline void add(int x,int y){
4704     tot++;des[tot] =y;
6d84     nxt[tot] = first[x];first[x] =tot;
95cf }
a4ef void tar(int node){
b081     dfn[node] = low[node] = ++dft;
5782     in[node] = 1;stk.push(node);
e83e     for (int t = first[node];t;t=nxt[t]){
e8e0         int v = des[t];
3c64         if (!dfn[v]){
53e9             tar(v);
9ee1             low[node] = min(low[node],low[v]);
8734         }else if (in[v]){
d1ad             low[node] = min(low[node],dfn[v]);
95cf         }
95cf     }
bb4b     if (dfn[node]==low[node]){
38ac         scc++;
1026         while (true){
6947             int temp = stk.top();
80c2             flag[temp]=scc;
5685             in[temp] = 0;
b820             cnt[scc]++;stk.pop();
ea28             if (temp==node)break;
95cf         }
95cf     }
95cf }
3117 int main(){
d994     scanf("%d%d%d",&n,&m,&h);
b8ca     for (int i=1;i<=n;i++){scanf("%d",t+i);}
356f     for (int i=0;i<m;i++){
4d1b         int u1,u2;scanf("%d%d",&u1,&u2);
7ec2         if (t[u1]==(t[u2]+1)%h)add(u2,u1);
e284         if (t[u2]==(t[u1]+1)%h)add(u1,u2);
95cf     }
6d72     for (int i=1;i<=n;i++){if (!dfn[i])tar(i);}
6dbf     for (int i=1;i<=n;i++){
f030         for (int t = first[i];t;t=nxt[t]){
f3e2             if (flag[i]==flag[des[t]])continue;
             else{d[flag[i]]++;}
         }
     }
     cnt[0] =n+1;int ans = 0;
     for (int i=1;i<=scc;i++){
         if (d[i]==0&&cnt[i]<cnt[ans]){ans = i;}
     }
     cout<<cnt[ans]<<endl;
     for (int i=1;i<=n;i++){
         if (flag[i]==ans){cout<<i<<"␣";}
     }
     cout<<endl;
     return 0;
}
```

a099 95cf 95cf 61a1 5176 83aa 95cf 31ae 6dbf e341 95cf 3251 7021 95cf

5.4 Dijkstra

```
// Created by calabash_boy on 18-11-13.
// remain k bi-edge such that the most points' dis == min_dis
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
const ll inf_ll = 0x3f3f3f3f3f3f3f3fll;
const int inf = 0x3f3f3f3f;
const int maxn = 300005;
struct EDGE{int first,second,third;};
int n,m,k;
namespace Short_Path_Tree{
    vector<pair<int,int> > Edge[maxn];
    bool used[maxn];
    void add_edge(int x,int y,int w) {Edge[x].push_back({y,w});}
    void output(const vector<int> &ans){
        printf("%d\n",(int) ans.size());
        for (int v : ans)printf("%d␣",v);
        puts("");exit(0);
    }
    void solve(int K){
        vector<int> ans(0);queue<int> Q;
        used[1] = 1;Q.push(1);
        while (!Q.empty()){
            if (ans.size()== K)output(ans);
            int head = Q.front();Q.pop();
```

427e 427e 302f 421c 4085 1c1d a7c7 8856 aaaa 47a0 04e9 db9e 727f b200 1e0b 90f7 69cb dcec 95cf 2fb6 8c27 2ad2 11e5 440f ff8a

```
            for (auto pr : Edge[head]){
                if (used[pr.first])continue;
                used[pr.first] = 1;
                ans.push_back(pr.second);
                Q.push(pr.first);
                if (ans.size()==K)output(ans);
            }
        }
        output(ans);
    }
};
namespace Dijkstra{
    ll dis[maxn];bool used[maxn];
    vector<EDGE > *Edge;int S,N;
    struct Node{
        int x;ll dis;
        bool operator < (const Node &other)const{
            return other.dis < dis;
        }
    };
    void init(vector<EDGE>*Edgee,int n,int st){
        Edge = Edgee;S =st;N = n;
    }
    void work(){
        memset(dis,inf,sizeof dis);
        priority_queue<Node> pq;
        dis[S] = 0;pq.push({S,0});
        while (!pq.empty()){
            Node head = pq.top();pq.pop();
            if (used[head.x])continue;
            used[head.x] = 1;
            for (auto pr : Edge[head.x]){
                if (dis[pr.first] > dis[head.x] + pr.second){
                    dis[pr.first] = dis[head.x] + pr.second;
                    pq.push({pr.first,dis[pr.first]});
                }
            }
        }
    }
    void extract_spt(){
        for (int u=1;u<=N;u++){
            for (auto pr : Edge[u]){
                if (dis[pr.first] == dis[u] + pr.second){
                    Short_Path_Tree::add_edge(u,pr.first,pr.third);
                }
            }
        }
    }
};
vector<EDGE> E[maxn];
int main(){
    scanf("%d%d%d",&n,&m,&k);
    for (int i=1;i<=m;i++){
        int x,y,w;scanf("%d%d%d",&x,&y,&w);
        E[x].push_back({y,w,i});
        E[y].push_back({x,w,i});
    }
    Dijkstra::init(E,n,1);
    Dijkstra::work();
    Dijkstra::extract_spt();
    Short_Path_Tree::solve(k);
    return 0;
}
```

5.5 Dijkstra interval graph

```
// CF 786B
#include<bits/stdc++.h>
using namespace std;
const int maxn = 1e5 + 100;
const int N = 10 * maxn;
typedef long long ll;
namespace Dijkstra{
    vector<pair<int,int> > E[N];
    ll dis[N];
    bool used[N];
    inline void add_edge(int u,int v,int w){
        E[u].push_back(make_pair(v,w));
    }
    void dijkstra(int S, int N){
        priority_queue<pair<ll,int> > pq;
        for (int i=1;i<=N;i++){
            dis[i] = 0x3f3f3f3f3f3f3f3fll;
            used[i] = 0;
        }
        dis[S] = 0;
```

```
cd0f        for (int i=1;i<=N;i++){
0f64            pq.push(make_pair(-dis[i],i));
95cf        }
57d6        while (!pq.empty()){
63ef            pair<ll,int> head = pq.top();pq.pop();
c89e            int u; ll dist;
4067            tie(dist,u) = head;
c884            dist *= -1;
9a95            if (used[u])continue;
db27            used[u] = 1;
48e2            for (auto e : E[u]){
33b3                int v,len;
ccc4                tie(v,len) = e;
f6e6                if (dis[v] > dist + len){
078a                    dis[v] = dist + len;
d06d                    pq.push(make_pair(-dis[v],v));
95cf                }
95cf            }
95cf        }
95cf    }
756f    void output(int n){
6dbf        for (int i=1;i<=n;i++){
b158            printf("%lld ",dis[i] == 0x3f3f3f3f3f3f3f3fll ? -1:dis[i]);
95cf        }
885d        puts("");
95cf    }
95cf}
24fc int n,q,s;
9f58 int cnt;
23cc struct SegmentTree{
c7e5    int id[maxn*4];
9476    void build(int x,int l,int r,bool up){
6281        id[x] = ++cnt;
3a0d        if (l == r){
c35b            int u = id[x];
d74c            int v = l;
2d00            if (up)swap(u,v);
a9ea            Dijkstra::add_edge(u, v, 0);
4f2d            return;
95cf        }
b8b7        int mid = l + r >> 1;
8094        build(x<<1,l,mid,up);
7d97        build(x<<1|1,mid+1,r,up);
c35b        int u = id[x];
dc32        int v = id[x<<1];
2d00        if (up)swap(u,v);
a9ea        Dijkstra::add_edge(u, v, 0);
a419        u = id[x];
e9c6        v = id[x<<1|1];
2d00        if (up)swap(u,v);
a9ea        Dijkstra::add_edge(u, v, 0);
95cf    }
3e8e    void add_edge(int x,int l,int r,int L,int R, int T, int w, bool up){
d499        if (l > R || L > r)return;
4d29        if (L <= l && r <= R){
c35b            int u = id[x];
8863            int v = T;
2d00            if (up)swap(u,v);
4c45            Dijkstra::add_edge(u, v, w);
4f2d            return;
95cf        }
b8b7        int mid = l + r >> 1;
9083        add_edge(x<<1, l, mid, L, R, T, w, up);
edd2        add_edge(x<<1|1, mid+1, r, L, R, T, w, up);
95cf    }
dfc9}Down,Up;
3117int main(){
13bb    scanf("%d%d%d",&n,&q,&s);
811f    cnt = n;
d237    Down.build(1, 1, n, false);
c1bc    Up.build(1, 1, n, true);
2cc8    while (q--){
aa14        int t,u,l,r,w;
8661        scanf("%d",&t);
8204        if (t == 1){
3b67            int v;
95a1            scanf("%d%d%d",&u,&v, &w);
8637            l = r = v;
96c0            t = 2;
8e2e        }else{
168f            scanf("%d%d%d%d",&u,&l,&r,&w);
95cf        }
163d        if (t == 2){
427e            // u -> [l,r], len = w
63b8            Down.add_edge(1, 1, n, l, r, u, w, true);
8e2e        }else{
427e            // [l,r] -> v, len = w
c4a7            Up.add_edge(1, 1, n, l, r, u, w, false);
```

```
95cf        }
95cf    }
3fd3    Dijkstra::dijkstra(s, cnt);
d041    Dijkstra::output(n);
7021    return 0;
95cf }
```

5.6 Eulor Tour

```
302f #include<bits/stdc++.h>
421c using namespace std;
52c1 const int maxn = 1e5 + 100;
a71b const int maxm = 5e5 + 100;
35b8 int n,m;
03f0 int d[maxn];
427e //<点，到这个点走的边id>
c49a vector<pair<int,int> > tour;
37e9 vector<pair<int,int> > E[maxn];
052c pair<int,int> edge[maxm];
f231 bool used[maxm];
880a int now[maxn];
5331 void dfs(int u,int e_id){
18c2     for (; now[u] < E[u].size(); now[u] ++){
6003         int v,id;
c7a3         tie(v,id) = E[u][now[u]];
1e6c         if (used[id]) continue;
6be5         used[id] = 1;
038b         dfs(v,id);
95cf     }
4556     tour.push_back(make_pair(u,e_id));
95cf }
3117 int main(){
ac98     scanf("%d%d",&n,&m);
e052     for (int i=1;i<=m;i++){
e635         int a,b;
a6b8         scanf("%d%d",&a,&b);
4a7b         edge[i] = make_pair(a,b);
7462         E[a].push_back(make_pair(b,i));
2a96         E[b].push_back(make_pair(a,i));
95cf     }
4e9d     dfs(1,-1);
8d42     reverse(tour.begin(), tour.end());
     /*
      for (auto pr : tour){
      int u,id;
      tie(u,id) = pr;
      cerr<<u<<" "<<id<<endl;
      }
      */
     return 0;
 }
```

87e7 3977 6b5b 2e37 6b68 95cf f2b5 7021 95cf

6 Graph/Tree

6.1 Divide & Conquer of Point

```
//
// Created by calabash_boy on 18-10-6.
//
//求树上长度小于等于k的有向路径数
#include<stdio.h>
#include<algorithm>
#include<cstring>
using namespace std;
const int MAX = 1e4+100;
const int INF = 0x3f3f3f3f;
int first [MAX*2];  int des[MAX*2];
int len[MAX*2];  int nxt[MAX*2];
int n,k,tot;  int a[MAX];  int sum[MAX];
int dp[MAX];  int dis[MAX];  int num,ans;
bool vis[MAX];  int Sum,Min,Minid;
void init(){
    memset(first,0,sizeof first);
    tot =0;  ans =0;
    memset(vis,0,sizeof vis);
}
inline void add(int x,int y,int z){
    tot++;
    des[tot]= y;  len[tot] =z;
    nxt[tot] = first[x];  first[x] = tot;
}
void input(){
    for (int i=1;i<n;i++){
        int u,v,w;
```

427e 427e 427e 427e 1915 54ff ef2f 421c bbaa 08a4 0b89 3efe 956f ecb3 aa8d 5d53 57d5 7ae1 87fb 95cf ce82 71cf 3615 6d84 95cf 0e91 324a 3676

```
95a1         scanf("%d%d%d",&u,&v,&w);
43a8         add(u,v,w);  add(v,u,w);
95cf     }
95cf }
da46 void dfs1(int node,int father){
90d3     sum[node] = 1;  dp[node] = 0;
e83e     for (int t = first[node];t;t = nxt[t]){
e8e0         int v = des[t];
c80a         if (v == father||vis[v]){
b333             continue;
95cf         }
d58d         dfs1(v,node);
cb59         sum[node] += sum[v];
2cf9         dp[node] = max(dp[node],sum[v]);
95cf     }
95cf }
2d8d void dfs2(int node,int father){
4ab1     int temp = max(dp[node],Sum-sum[node]);
d6e3     if (temp<Min){
76f6         Min = temp;  Minid = node;
95cf     }
e83e     for (int t = first[node];t;t = nxt[t]){
e8e0         int v = des[t];
a37f         if (v==father||vis[v]){  continue;  }
253c         dfs2(v,node);
95cf     }
95cf }
6fae int getRoot(int u){
8e67     dfs1(u,0);  Sum = sum[u];
3069     Min = INF;  Minid = -1;
005f     dfs2(u,0);
1090     return Minid;
95cf }
4ac1 void getDist(int node,int father,int dist){
e097     dis[num++] = dist;
e83e     for (int t = first[node];t;t = nxt[t]){
e8e0         int v =des[t];
a37f         if (v == father||vis[v]){  continue;  }
6cae         getDist(v,node,dist+len[t]);
95cf     }
95cf }
97e3 int calc (int u,int val){
9daa     num=0;  int res =0;
d05a     getDist(u,0,0);
4b02     sort(dis,dis+num);
e78d     int i=0;int j=num-1;
6f80     while (i<j){
e6c0         if (dis[i]+dis[j]+2*val<=k){
efef             res+=j-i;
a42b             i++;
5cd2         }else{  j--;  }
95cf     }
244d     return res;
95cf }
ee28 void solve(int u){
b583     int root = getRoot(u);
b2e3     ans +=calc(root,0);  vis[root] = true;
235c     for (int t = first[root];t;t = nxt[t]){
e8e0         int v = des[t];
332f         if (vis[v]){
b333             continue;
95cf         }
91fa         ans-=calc(v,len[t]);
a707         solve(v);
95cf     }
95cf }
3117 int main(){
7666     while (scanf("%d%d",&n,&k)!=EOF&&n&&k){
07e2         init();
2a5c         input();
1d60         solve(1);
53b1         printf("%d\n",ans);
95cf     }
7021     return 0;
95cf }
```

6.2 Heavy Light Decomposition

```
427e // Created by calabash_boy on 18-7-3.
427e //统计路径上标记边的个数
302f #include<bits/stdc++.h>
421c using namespace std;
8e62 const int maxn = 500000+100;
4bc9 int n,q,m,Root;  char s[10];
5f7d struct BIT{
3bf5     int sm[maxn];
```

```
cf5a     int lowbit(int _x){return _x&(-_x);}
d5af     void build (int l,int r){
5023         for (int i=l;i<=r;i++)add(i,1);
95cf     }
6142     void add(int x,int val){
dc9a         while (x<=maxn){
9ccc             sm[x]+=val;x+=lowbit(x);
95cf         }
95cf     }
eb61     int sum(int x){
5839         int res =0;
6f1c         while (x){
e64f             res+=sm[x];
e6b6             x-=lowbit(x);
95cf         }
244d         return res;
95cf     }
9fc7     int query_sum(int l,int r){
7789         return sum(r)-sum(l-1);
95cf     }
b0c1 }tree;
9c21 namespace Heavy_Light_Decomposition{
7b14     int first[maxn*2];int nxt[maxn*2];int des[maxn*2];
cd30     int tot,cnt=0;
0d93     int tpos[maxn];int dep[maxn];int top[maxn];
d6bf     int fa[maxn]; int wson[maxn];  int sz[maxn];
f9d3     inline void addEdge(int _u, int _v){
26b9         des[++tot] = _v;
a66a         nxt[tot] = first[_u];
593b         first[_u] = tot;
95cf     }
427e     //统计dep，子树sz，重儿子wson
dd7c     void dfs(int node,int father){
c5b1         dep[node] = dep[father]+1;
afa3         fa[node] = father;  sz[node] =1;
e83e         for (int t = first[node];t;t = nxt[t]){
e8e0             int v = des[t];
e092             if (v==father){  continue;  }
1f8e             dfs(v,node);
acb3             if (sz[v]>sz[wson[node]]){
44c0                 wson[node] = v;
95cf             }
47d5             sz[node]+=sz[v];
95cf         }
95cf     }
427e     //node所在链的头是chain
aee5     void dfs2(int node,int father,int chain){
950f         top[node] = chain;  tpos[node] = ++cnt;
d010         if (wson[node]){
0f73             dfs2(wson[node],node,chain);
95cf         }
e83e         for (int t = first[node];t;t = nxt[t]){
e8e0             int v = des[t];
b928             if (v==father||v ==wson[node]){  continue;  }
e6aa             dfs2(v,node,v);
95cf         }
95cf     }
c352     /* s 树根 */
1a86     void init(int root){
5136         dfs(root,0);
7cdf         dfs2(root, 0, root);
95cf     }
620b     int lca(int x,int y){
d2f8         while (top[x]!=top[y]){
0cc5             if (dep[top[x]]<dep[top[y]]){swap(x,y);}
7456             x = fa[top[x]];
95cf         }
d22b         if (dep[x]<dep[y])swap(x,y);
c218         return y;
95cf     }
29cf     void modify(int u,int v){
733e         if (fa[u]!=v){  swap(u,v);  }
1e27         tree.add(tpos[u],-1);
95cf     }
1dc2     int get_sum(int u,int v){
5839         int res =0;
03a1         while (top[u]!=top[v]){
a716             if (dep[top[u]]<dep[top[v]]){  swap(u,v);  }
f1e8             res+= tree.query_sum(tpos[top[u]],tpos[u]);
005b             u = fa[top[u]];
95cf         }
4b1a         if (dep[u]<dep[v]){  swap(u,v); }
cbff         res += tree.query_sum(tpos[v],tpos[u]);
244d         return res;
95cf     }
95cf }
3117 int main(){
cd91     scanf("%d",&n);
```

```
324a    for (int i=1;i<n;i++){
17be        int u,v;  scanf("%d%d",&u,&v);
1478        Heavy_Light_Decomposition::addEdge(u, v);
e4e6        Heavy_Light_Decomposition::addEdge(v, u);
95cf    }
90e1    Heavy_Light_Decomposition::init(1);
427e    //维护
1ca5    tree.build(2,n);
ea85    scanf("%d",&q);
3605    q+=n-1;
2cc8    while (q--){
587c        scanf("%s",s);
5d10        if (s[0]=='W'){
3c9e            int x;
ea4e            scanf("%d",&x);
3b50            printf("%d\n",Heavy_Light_Decomposition::get_sum(1,x));
8e2e        }else{
0f8b            int x,y;
a9b3            scanf("%d%d",&x,&y);
a309            Heavy_Light_Decomposition::modify(x,y);
95cf        }
95cf    }
7021    return 0;
95cf}
```

6.3 Virtual Tree

```
427e//
427e// Created by calabash_boy on 18-10-6.
427e//
427e
302f#include <bits/stdc++.h>
421cusing namespace std;
5cadtypedef long long LL;
40fbconst int maxn = 25e4+100;
b1ecconst LL INF = 0x3f3f3f3f3f3f3f3fLL;
58a9int first[maxn],des[maxn*2],nxt[maxn*2],tot;
35b8int n,m;
667aLL dp[maxn],leng[maxn*2], len[maxn];
e55bint vis[maxn],dep[maxn],fa[maxn];
21feint sz[maxn],wson[maxn],ttop[maxn],tfa[maxn];int k,h[maxn];
0a19int stk[maxn],top;int l[maxn],r[maxn],dfs_clock;
```

```
a50ainline void addEdge(int x,int y,int w){
71cf    tot++;
a752    des[tot] = y;leng[tot] = w;
6d84    nxt[tot] = first[x];first[x] = tot;
95cf}
827dvoid dfs(int u,int fath){
84cf    l[u] = ++dfs_clock;sz[u]=1;
3ddf    for (int t = first[u];t;t=nxt[t]){
e8e0        int v = des[t];
9d74        if (v==fath)continue;
62a8        LL w = leng[t];
e4a6        dep[v] = dep[u] + 1;tfa[v]=u;
818a        len[v] = min(len[u],w);
7457        dfs(v,u);sz[u]+=sz[v];
c7eb        if (sz[v]>sz[wson[u]]){wson[u] = v;}
95cf    }
f142    r[u]=dfs_clock ;
95cf}
4707void dfs2(int u,int chain){
0865    ttop[u]=chain;
d6b4    if (wson[u])dfs2(wson[u],chain);
3ddf    for (int t = first[u];t;t=nxt[t]){
e8e0        int v = des[t];
0c51        if (v==tfa[u]||v==wson[u])continue;
8064        dfs2(v,v);
95cf    }
95cf}
620bint lca(int x,int y){
00da    while (ttop[x]!=ttop[y]){
6d86        if (dep[ttop[x]]<dep[ttop[y]])swap(x,y);
2df6        x = tfa[ttop[x]];
95cf    }
d22b    if (dep[x]<dep[y])swap(x,y);
c218    return y;
95cf}
4ac9bool cmp(int x,int y){return l[x]<l[y];}
9627void solve(){
c93a    scanf("%d",&k);
f3ea    for (int i=0;i<k;i++){
3596        scanf("%d",h+i);
a234        vis[h[i]]=1;dp[h[i]]=0;
95cf    }
f5bb    sort(h,h+k,cmp);
a555    int kk =k;
```

```
c701     for (int i=1;i<kk;i++){
4680         int temp = lca(h[i-1],h[i]);
b925         if (!vis[temp])vis[temp]=2,h[k++] =temp,dp[temp]=0;
95cf     }
22a9     if (!vis[1])vis[1]=2,h[k++]=1,dp[1]=0;
f5bb     sort(h,h+k,cmp);
25a6     top=1;stk[0]=h[0];
3ef4     for (int i=1;i<k;i++){
b35a         while (l[h[i]]>r[stk[top-1]])top--;
f930         fa[h[i]] = stk[top-1];
274e         stk[top++] =h[i];
95cf     }
5c52     for (int i=k-1;i>=0;i--){
dca2         if (vis[h[i]]==2)dp[h[i]] = min(dp[h[i]],len[h[i]]);
6a6b         else dp[h[i]] = len[h[i]];
d6ae         dp[fa[h[i]]]+=dp[h[i]];
95cf     }
c682     printf("%lld\n",dp[1]);
f3ea     for (int i=0;i<k;i++){
e3ec         vis[h[i]]=0;
95cf     }
95cf }
3117 int main(){
cd91     scanf("%d",&n);
324a     for (int i=1;i<n;i++){
3676         int u,v,w;
95a1         scanf("%d%d%d",&u,&v,&w);
8796         addEdge(u,v,w);addEdge(v,u,w);
95cf     }
8694     len[0] = len[1] = INF;
0e9e     dfs(1,-1);dfs2(1,1);
aa8d     scanf("%d",&m);
74ed     while (m--){solve();}
7021     return 0;
95cf }
```

7 Math

7.1 FFT

```
427e // Created by calabash_boy on 18-6-18.
302f #include <bits/stdc++.h>
```

```
421c using namespace std;
e48c namespace fft {
427e     //attention data type
53f7     typedef long long type;
f7dc     typedef double db;
e718     struct cp {
ba04         db x, y;
cfb3         cp() { x = y = 0; }
f329         cp(db x, db y) : x(x), y(y) {}
329b     };
9f2f     cp operator+(cp a, cp b) { return cp(a.x + b.x, a.y + b.y); }
624b     cp operator-(cp a, cp b) { return cp(a.x - b.x, a.y - b.y); }
36fe     cp operator*(cp a, cp b) { return cp(a.x * b.x - a.y * b.y, a.x * b.y + a.y
           * b.x); }
a0e1     cp conj(cp a) { return cp(a.x, -a.y); }
6ecb     type base = 1;
44b9     vector<cp> roots = {{0, 0}, {1, 0}};
3a50     vector<type> rev = {0, 1};
3f9e     const db PI = acosl(-1.0);
2b5b     void ensure_base(type nbase) {
7037         if (nbase <= base) return;
bbb1         rev.resize(static_cast<unsigned long>(1 << nbase));
89c3         for (type i = 0; i < (1 << nbase); i++) {
33a9             rev[i] = (rev[i >> 1] >> 1) + ((i & 1) << (nbase - 1));
95cf         }
a0ef         roots.resize(static_cast<unsigned long>(1 << nbase));
7acf         while (base < nbase) {
cd10             db angle = 2 * PI / (1 << (base + 1));
f864             for (type i = 1 << (base - 1); i < (1 << base); i++) {
b824                 roots[i << 1] = roots[i];
90ee                 db angle_i = angle * (2 * i + 1 - (1 << base));
a5d7                 roots[(i << 1) + 1] = cp(cos(angle_i), sin(angle_i));
95cf             }
d27a             base++;
95cf         }
95cf     }
3548     void fft(vector<cp> &a, type n = -1) {
805a         if (n == -1) n = a.size();
2fa3         assert((n & (n - 1)) == 0);
dca5         type zeros = __builtin_ctz(n);
c44f         ensure_base(zeros);
a1b9         type shift = base - zeros;
800c         for (type i = 0; i < n; i++) {
aa3c             if (i < (rev[i] >> shift)) {
```

```
669c                swap(a[i], a[rev[i] >> shift]);
95cf            }
95cf        }
5911        for (type k = 1; k < n; k <<= 1) {
b660            for (type i = 0; i < n; i += 2 * k) {
b247                for (type j = 0; j < k; j++) {
7dca                    cp z = a[i + j + k] * roots[j + k];
ee2d                    a[i + j + k] = a[i + j] - z;
4da7                    a[i + j] = a[i + j] + z;
95cf                }
95cf            }
95cf        }
95cf    }
fbc2    vector<cp> fa, fb;
6833    vector<type> multiply(vector<type> &a, vector<type> &b) {
02f0        type need = a.size() + b.size() - 1;
cf09        type nbase = 0;
0c88        while ((1 << nbase) < need) nbase++;
6f7d        ensure_base(nbase);
cb07        type sz = 1 << nbase;
b44d        if (sz > (type) fa.size())
74d8            fa.resize(static_cast<unsigned long>(sz));
46e8        for (type i = 0; i < sz; i++) {
2155            type x = (i < (type) a.size() ? a[i] : 0);
f2d7            type y = (i < (type) b.size() ? b[i] : 0);
140d            fa[i] = cp(x, y);
95cf        }
eb13        fft(fa, sz);
53b1        cp r(0, -0.25 / sz);
6611        for (type i = 0; i <= (sz >> 1); i++) {
3695            type j = (sz - i) & (sz - 1);
f17e            cp z = (fa[j] * fa[j] - conj(fa[i] * fa[i])) * r;
4a23            if (i != j) {
0628                fa[j] = (fa[i] * fa[i] - conj(fa[j] * fa[j])) * r;
95cf            }
8cd4            fa[i] = z;
95cf        }
eb13        fft(fa, sz);
a834        vector<type> res(static_cast<unsigned long>(need));
4516        for (type i = 0; i < need; i++) {
1653            res[i] = fa[i].x + 0.5;
95cf        }
244d        return res;
95cf    }
```

```
3ca7    vector<type> multiply_mod(vector<type> &a, vector<type> &b, type m, type eq
          = 0) {
02f0        type need = a.size() + b.size() - 1;
cf09        type nbase = 0;
0c88        while ((1 << nbase) < need) nbase++;
6f7d        ensure_base(nbase);
cb07        type sz = 1 << nbase;
3292        if (sz > (type) fa.size()) {
74d8            fa.resize(static_cast<unsigned long>(sz));
95cf        }
2f67        for (type i = 0; i < (type) a.size(); i++) {
cfe6            type x = (a[i] % m + m) % m;
7cb0            fa[i] = cp(x & ((1 << 15) - 1), x >> 15);
95cf        }
b1cb        fill(fa.begin() + a.size(), fa.begin() + sz, cp {0, 0});
eb13        fft(fa, sz);
8c71        if (sz > (type) fb.size()) {
14b9            fb.resize(static_cast<unsigned long>(sz));
95cf        }
2cba        if (eq) {
88c2            copy(fa.begin(), fa.begin() + sz, fb.begin());
8e2e        } else {
0ac2            for (type i = 0; i < (type) b.size(); i++) {
ad83                type x = (b[i] % m + m) % m;
97f9                fb[i] = cp(x & ((1 << 15) - 1), x >> 15);
95cf            }
5f8e            fill(fb.begin() + b.size(), fb.begin() + sz, cp {0, 0});
e06b            fft(fb, sz);
95cf        }
d8f2        db ratio = 0.25 / sz;
9cc7        cp r2(0, -1);cp r3(ratio, 0);
0367        cp r4(0, -ratio);cp r5(0, 1);
6611        for (type i = 0; i <= (sz >> 1); i++) {
3695            type j = (sz - i) & (sz - 1);
996e            cp a1 = (fa[i] + conj(fa[j]));
a37e            cp a2 = (fa[i] - conj(fa[j])) * r2;
51fd            cp b1 = (fb[i] + conj(fb[j])) * r3;
ad90            cp b2 = (fb[i] - conj(fb[j])) * r4;
4a23            if (i != j) {
792b                cp c1 = (fa[j] + conj(fa[i]));
ecde                cp c2 = (fa[j] - conj(fa[i])) * r2;
18a0                cp d1 = (fb[j] + conj(fb[i])) * r3;
6ced                cp d2 = (fb[j] - conj(fb[i])) * r4;
28c4                fa[i] = c1 * d1 + c2 * d2 * r5;
```

```
178d                fb[i] = c1 * d2 + c2 * d1;
95cf            }
1184            fa[j] = a1 * b1 + a2 * b2 * r5;
87e9            fb[j] = a1 * b2 + a2 * b1;
95cf        }
922b        fft(fa, sz);fft(fb, sz);
a834        vector<type> res(static_cast<unsigned long>(need));
4516        for (type i = 0; i < need; i++) {
9dbc            long long aa = fa[i].x + 0.5;
d335            long long bb = fb[i].x + 0.5;
de5d            long long cc = fa[i].y + 0.5;
67e4            res[i] = (aa + ((bb % m) << 15) + ((cc % m) << 30)) % m;
95cf        }
244d        return res;
95cf    }
2307    vector<type> square_mod(vector<type> &a, type m) {
b845        return multiply_mod(a, a, m, 1);
95cf    }
329b };
eb45 const int maxn = 2e5+100;
86d1 int n,x;
7608 int a[maxn],sum[maxn],cnt[maxn];
a6aa vector<long long > A,B,C;
427e //example:
427e //f[i] = number of subsequences whose occurence of 1 is i.
427e //f[i] = \sum_{cnt[j]*cnt[j-i]}
3117 int main(){
a5fe    scanf("%d%d",&n,&x);cnt[0]=1;
6dbf    for (int i=1;i<=n;i++){
60cb        scanf("%d",a+i);
9a8f        sum[i] =sum[i-1];
1229        if(a[i]<x)sum[i]++;
6210        cnt[sum[i]]++;
95cf    }
bb11    A.resize(n*2+2);B.resize(n*2+2);
0423    for (int i=0;i<=n;i++){
1451        A[n+i] = cnt[i];B[n-i] = cnt[i];
95cf    }
284a    C = fft::multiply(A,B);
7cf7    C[n*2]-=n+1;C[n*2]>>=1;
d7c0    for (int i=n*2;i<=n*3;i++){ cout<<C[i]<<"␣"; }
7021    return 0;
95cf }
```

7.2 FWT

```
427e // Created by calabash_boy on 18-8-17.
427e //UOJ 310
302f #include<bits/stdc++.h>
421c using namespace std;
5cad typedef long long LL;
a923 const int N = 1048576;;
5bf2 const int MOD = 998244353;
2003 const int INV2 = (MOD+1)>>1;
4d4d const int INV4 = 1LL*INV2*INV2%MOD;
ac9d int a[N];
5c83 int n;
427e //xor fwt : A[i] = \sigma{-1^([i&j])*a[j]}   [x]:count of 1-bit
3284 void FWT(int *a,int n,int r){
65de     for (int i=1;i<n;i<<=1){
2d6f         for (int j=0;j<n;j+=(i<<1)){
3d77             for (int k =0;k<i;k++){
269d                 int x = a[j+k];int y = a[j+k+i];
f418                 if (r){
a62b                     a[j+k] = (x+y)%MOD;
df0f                     a[j+k+i] = (x-y+MOD)%MOD;
8e2e                 }else{
a36d                     a[j+k] = 1LL*(x+y)*INV2%MOD;
5b23                     a[j+k+i] = 1LL*(x-y+MOD)*INV2%MOD;
95cf                 }
95cf             }
95cf         }
95cf     }
95cf }
e854 LL pow_mod(LL x,LL y){
1938     LL ret = 1;
4fc6     for (;y;y>>=1){if (y&1) ret = ret*x%MOD;x = x*x%MOD;}
ee0f     return ret;
95cf }
3117 int main(){
cd91     scanf("%d",&n);
6dbf     for (int i=1;i<=n;i++){
7681         int x;scanf("%d",&x);
52fe         a[x]++;
95cf     }
564e     FWT(a,N,1);
8cc2     for(int i=0;i<N;i++){
788a         a[i] = (n+2*a[i])%MOD;
```

```
2be0        int cnt3 = 1LL*(a[i]+n)%MOD*INV4%MOD;
c3f6        int cnt1 = n-cnt3;
557b        a[i] = pow_mod(3,cnt3);
9f4a        if (cnt1&1)a[i] = MOD-a[i];
95cf    }
e16f    FWT(a,N,0);
369d    printf("%d\n",(a[0]+MOD-1)%MOD);
7021    return 0;
95cf}
```

7.3 BerlekampMassey

```
427e// Created by calabash_boy on 18-8-16.
302f#include<bits/stdc++.h>
d196#define FOR(i,l,r) for (int i = (l);i<(r);i++)
ba3e#define FORD(i,r,l) for (int i= (r);i>(l);i--)
421cusing namespace std;
5cadtypedef long long LL;
7c77typedef vector<LL> V;
b575const int MOD = 1e9+7;
427e// k 为 m 最高次数 且  a[m] == 1
70d2namespace BerlekampMassey {
a44f    inline void up(LL& a, LL b) { (a += b) %= MOD; }
427e
68c4    V mul(const V& a, const V& b, const V& m, int k) {
138d        V r; r.resize(2 * k - 1);
4c60        FOR (i, 0, k)
d87c            FOR (j, 0, k)
01e3                up(r[i + j], a[i] * b[j]);
43e8        FORD (i, k - 2, -1) {
d87c            FOR (j, 0, k)
bbda                up(r[i + j], r[i + k] * m[j]);
57fc            r.pop_back();
95cf        }
547e        return r;
95cf    }
e854    LL pow_mod (LL x,LL y){
1938        LL ret =1;
4fc6        for (;y;y>>=1){if (y&1) ret = ret*x%MOD;x = x * x %MOD;}
ee0f        return ret;
95cf    }
025b    LL get_inv(LL x,LL MOD){
a4c6        return pow_mod(x,MOD-2);
95cf    }
b35e    V pow(LL n, const V& m) {
737d        int k = (int)m.size() - 1; assert(m[k] == -1 || m[k] == MOD - 1);
bd5c        V r(k), x(k); r[0] = x[1] = 1;
ddfe        for (; n; n >>= 1, x = mul(x, x, m, k))
77c0            if (n & 1) r = mul(x, r, m, k);
547e        return r;
95cf    }
0d21    LL go(const V& a, const V& x, LL n) {
427e        // a: (-1, a1, a2, ..., ak).reverse
427e        // x: x1, x2, ..., xk
427e        // x[n] = sum[a[i]*x[n-i],{i,1,k}]
84ec        int k = (int)a.size() - 1;
f0f5        if (n <= k) return x[n - 1];
4690        V r = pow(n - 1, a);
f7ff        LL ans = 0;
4c60        FOR (i, 0, k)
d862            up(ans, r[i] * x[i]);
4206        return ans;
95cf    }
427e
ad3d    V BM(const V& x) {
89e6        V a = {-1}, b = {233};
c493        FOR (i, 1, x.size()) {
73f7            b.push_back(0);
6453            LL d = 0, la = a.size(), lb = b.size();
d228            FOR (j, 0, la) up(d, a[j] * x[i - la  + 1 + j]);
85ae            if (d == 0) continue;
292f            V t; for (auto& v: b) t.push_back(d * v % MOD);
296a            FOR (j, 0, a.size()) up(t[lb - 1 - j], a[la - 1 - j]);
3ead            if (lb > la) {
46e5                b = a;
f0ce                LL inv = -get_inv(d, MOD);
b92f                for (auto& v: b) v = v * inv % MOD;
95cf            }
64bf            a.swap(t);
95cf        }
b24a        for (auto& v: a) up(v, MOD);
5ffd        return a;
95cf    }
bb1a    void sample();
95cf}
f425void BerlekampMassey::sample(){
```

```
3ddb     V x(6);
26b0     x[0] = 1;x[1] = 2;
dc7c     x[2] = 21;x[3] = 212;
408c     x[4] = 2141;x[5] = 21622;
6243     V a = BerlekampMassey::BM(x);
a849     cout<<"a[n]=";
0126     for (int i = 0;i<a.size()-2;i++){
844c         cout<<a[i]<<"*a[n-"<<a.size()-1-i<<"]+";
95cf     }
e0ba     cout<<a[a.size()-2]<<"*a[n-1]"<<endl;
95cf }
3117 int main(){
47ff     BerlekampMassey::sample();
7021     return 0;
95cf }
```

7.4 CRT

```
427e //
427e // Created by DELL on 2019/2/12.
427e //luogu 4777
302f #include<bits/stdc++.h>
421c using namespace std;
4085 typedef long long ll;
52c1 const int maxn = 1e5+100;
ff57 namespace CRT{
8345     ll ex_gcd(ll a,ll b,ll& x,ll& y){
7d1a         if (b == 0){x = 1;y = 0;return a;}
df10         ll gcd = ex_gcd(b,a%b,x,y);
8737         ll t = x;x = y;y = t - a/b*y;
8be6         return gcd;
95cf     }
40a5     ll mul_mod(ll a,ll b,ll m){
292f         ll res = 0;
ca22         while (b){
90a9             if (b&1){
6d81                 res = (res + a) % m;
95cf             }
ca1f             b >>=1;
06e5             a = a * 2 % m;
95cf         }
244d         return res;
95cf     }
427e     // ans = first + t * second;
427e     // x = second (mod first)
7f60     pair<ll,ll>work(vector<pair<ll,ll> >&es ){
601c         ll ans = es[0].second;
2a60         ll M = es[0].first;
954a         for (int i=1;i<es.size();i++){
c35f             ll a = es[i].first;
27e2             ll b = es[i].second;
d406             ll x,y;
6786             ll gcd = ex_gcd(M,a,x,y);
69fb             ll c = (b - ans %a + a) % a;
1a20             a/=gcd;
e23e             if (c % gcd)return {-1,-1};
5a47             x = (mul_mod(x ,(c / gcd),a) + a)% a;
4108             ans += M * x;
9b2a             M *= a;
324d             ans %= M;
95cf         }
f267         return {ans,M};
95cf     }
95cf }
6a81 vector<pair<ll,ll> > es;
3117 int main(){
5c83     int n;
cd91     scanf("%d",&n);
1294     for (int i=0;i<n;i++){
6d1c         ll a,b;
9407         scanf("%lld%lld",&a,&b);
3a4a         es.push_back(make_pair(a,b));
95cf     }
c88b     pair<ll,ll> ans = CRT::work(es);
427e    // cout<<ans.first<<" "<<ans.second<<endl;
ee13     ll x = ans.first;
290b     cout<<x<<endl;
7021     return 0;
95cf }
```

7.5 Linear Sieve

```
302f #include<bits/stdc++.h>
421c using namespace std;
```

```
68e4 const int maxn = 1e7+10;
4085 typedef long long ll;
727f bool used[maxn];
efe5 int mu[maxn];
7c8f vector<int> prime;
c882 ll f[maxn];
a0b1 int low[maxn];
22c5 void sieve(int size){
427e     //f:multiplicative function;
7d97     assert(size < maxn);
7f5a     mu[1] = 1;
c6b9     f[1] = 1;
40bd     for (int i=2;i<=size;i++){
efb1         if (!used[i]){
1024             prime.push_back(i);
7171             mu[i] = -1;
427e             //f:TODO
c21b             low[i] = i;
95cf         }
eb1a         for (int j = 0;j < prime.size();j++){
d3c2             ll nxt = 1ll * i * prime[j];
b561             if (nxt > size)break;
6b89             used[nxt] = 1;
073a             if (i % prime[j]){
b9b8                 low[nxt] = prime[j];
66f9                 mu[nxt] = -mu[i];
427e                 //f: mod or not?
7225                 f[nxt] = f[i] * f[prime[j]];
8e2e             }else{
734b                 low[nxt] = prime[j] * low[i];
8ec3                 mu[nxt] = 0;
b401                 if (low[nxt] != nxt){
427e                     //mod or not?
4d18                     f[nxt] = 1ll * f[low[nxt]] * f[nxt/low[nxt]];
8e2e                 }else{
427e                     // i = prime[j] ^ k
427e                     //f:TODO
95cf                 }
6173                 break;
95cf             }
95cf         }
95cf     }
95cf }
3117 int main(){
ff91     sieve(1e7);
7021     return 0;
95cf }
```

7.6 Linear Basis

```
6c13 /* Generated by powerful Codeforces Tool
c7a5  * Author: calabash_boy_love_15
6619  * Time: 2019-05-15 11:00:02
ca63  * Personal Code Template: https://github.com/4thcalabash/ACM-Code-Library
421d **/
302f #include <bits/stdc++.h>
421c using namespace std;
4c95 int s[maxn];
5c83 int n;
2360 struct Linear_Basis{
427e     //basis vector
d2e8     int basis[22];
427e     //basis vector in origin data
36c3     int num[22];
1126     void clear(){
037d         memset(basis,0,sizeof basis);
7b40         memset(num,0,sizeof num);
95cf     }
2f9f     void ins(int x){
c7a6         int bk = x;
54c0         for (int i=20;i>=0;i--){
a0f3             if (x & (1<< i)){
e222                 if (!basis[i]){basis[i] = x;num[i] = bk;break;}
370c                 x ^= basis[i];
95cf             }
95cf         }
95cf     }
5bcc     int count(){
8abb         int cnt = 0;
9f1c         for (int i=0;i<=20;i++){
340e             cnt += (basis[i] != 0);
95cf         }
6808         return cnt;
95cf     }
56dd     void debug(){
af23         _debug("basis : ");
```

```
9f1c        for (int i=0;i<=20;i++){
dbf5            if (basis[i])_debug("%d␣:␣%d",i,basis[i]);
95cf        }
95cf    }
4a42 }basis;
3117 int main(){
e1b6    cin>>n;
6dbf    for (int i=1;i<=n;i++){
f9af        cin>>s[i];
9f1c        basis.ins(s[i]);
95cf    }
7021    return 0;
95cf }
```

7.7 Matrix

```
302f #include <bits/stdc++.h>
421c using namespace std;
582c const double EPS = 1e-18;
5480 template<class Type>
47d5 inline bool is_zero(Type value){
1088   return fabs(value) <= EPS;
95cf }
427e
5480 template<class Type>
f717 class Matrix{
33f9 private:
d7e1   vector<vector<Type> > data;
63d4 public:
06a1   int width,height;
d7bf   Matrix(int height=0,int width=0,Type value = 0);
f71d   Matrix<Type> (const Matrix<Type> & other);
c663   Matrix<Type> operator + (const Matrix<Type> & other);
4970   Matrix<Type> operator - (const Matrix<Type> & other);
05bc   Matrix<Type> operator * (const Matrix<Type> & other);
ac53   Matrix<Type> operator ~();
78dd   vector<Type> operator [](int row)const;
79fa   vector<Type>& operator [] (int row);
92d1   void print();
e53f   static Matrix<Type> eye(int n);
329b };
598c typedef Matrix<double> Mat;
```

```
5480 template<class Type>
b1fb Matrix<Type>::Matrix(const Matrix<Type> & other){
ec94   height = other.height;
4825   width = other.width;
af45   data = other.data;
95cf }
5480 template<class Type>
159a Matrix<Type>::Matrix(int height_,int width_,Type value_){
b275   height = height_;
7c4b   width = width_;
0a0c   data.resize(height);
b487   for (int i=0;i< height;i++){
2d2a     data[i].resize(width,value_);
95cf   }
95cf }
5480 template<class Type>
6d0a void Matrix<Type>::print(){
b487   for (int i=0;i<height;i++){
8c04     for (int j=0;j< width;j++){
dc25       cout<<data[i][j]<<"␣";
95cf     }
3251     cout<<endl;
95cf   }
95cf }
5480 template<class Type>
3d0f Matrix<Type> Matrix<Type> :: operator + (const Matrix <Type> & other){
5f42   if (other.height != height || other.width != width){
70ac     throw -1;
95cf   }
621e   Matrix<Type> res(height,width);
b487   for (int i=0;i< height;i++){
8c04     for (int j=0;j< width;j++){
2b5a       res.data[i][j] = data[i][j] + other.data[i][j];
95cf     }
95cf   }
244d   return res;
95cf }
5480 template<class Type>
dba8 Matrix<Type> Matrix<Type> :: operator - (const Matrix<Type> & other){
5f42   if (other.height != height || other.width != width){
70ac     throw -1;
95cf   }
621e   Matrix<Type> res(height,width);
b487   for (int i=0;i< height;i++){
```

```
8c04    for (int j=0;j< width;j++){
bf9d      res.data[i][j] = data[i][j] - other.data[i][j];
95cf    }
95cf  }
244d  return res;
95cf }
5480 template<class Type>
fd48 Matrix<Type> Matrix<Type> :: operator * (const Matrix<Type> & other){
3007  if ( other.height != width){
e3f5    throw -2;
95cf  }
a271  Matrix<Type> res(height,other.width);
b487  for (int i=0;i< height;i++){
e971    for (int j=0;j< other.width;j++){
f940      for (int k=0;k<width;k++){
5ee4        res.data[i][j] += data[i][k] * other.data[k][j];
95cf      }
95cf    }
95cf  }
244d  return res;
95cf }
5480 template<class Type>
e456 Matrix<Type> Matrix<Type>:: operator ~(){
354a  int h = height;
d78c  int w = width;
1328  Matrix<Type> res(w,h);
3659  for (int i=0;i<width;i++){
eddd    for (int j=0;j<height;j++){
aeae      res[i][j] = data[j][i];
95cf    }
95cf  }
244d  return res;
95cf }
5480 template<class Type>
7540 vector<Type> Matrix<Type> :: operator[](int row)const{
0ba7  cout<<row<<"␣"<<height<<endl;
3f38  if (row > height){
6ffd    throw -5;
95cf  }
701d  return data[row];
95cf }
5480 template<class Type>
1ec7 vector<Type>& Matrix<Type> :: operator[](int row){
3f38  if (row > height){
6ffd    throw -5;
95cf  }
701d  return data[row];
95cf }
5480 template<class Type>
31a4 Matrix<Type> Matrix<Type> :: eye(int n){
d659  Matrix<Type> res(n,n);
1294  for (int i=0;i<n;i++){
a2e5    res[i][i] = 1;
95cf  }
244d  return res;
95cf }
3117 int main(){
c6a7    Mat test(3,5,2.0);
f07b    test.print();
7021    return 0;
95cf }
```

7.8 Mobius

```
e9ac /* x in [1,N]; y in [1,M] (x,y) = 1 */
59b9 #include<cstdio>
09f7 #include<vector>
421c using namespace std;
52c1 const int maxn = 1e5+100;
4085 typedef long long ll;
727f bool used[maxn];
7c8f vector<int> prime;
a00a ll mu[maxn];
9bc6 void sieve(){
7f5a    mu[1] = 1;
82c4    for (int i=2;i<maxn;i++){
efb1        if(!used[i]){
1024            prime.push_back(i);
7171            mu[i] = -1;
95cf        }
eb1a        for (int j = 0;j<prime.size();j++){
b70b            long long nxt = 1ll* prime[j] * i;
1487            if(nxt >= maxn)break;
6b89            used[nxt] = 1;
20cc            if (i % prime[j] == 0){
8ec3                mu[nxt] = 0;
```

```
6173                break;
8e2e            }else{
66f9                mu[nxt] = -mu[i];
95cf            }
95cf        }
95cf    }
95cf}
8399ll work(int n,int m){
19f3    ll ans = 0;
78fb    int top = min(n,m);
3d1c    for (int i=1;i<=top;i++){
7d55        ans += 1ll * mu[i] * (n/i) * (m/i);
95cf    }
4206    return ans;
95cf}
3117int main(){
5ec4    sieve();
9523    int T;
1fd9    scanf("%d",&T);
9415    for (int Case = 1;Case <= T;Case ++){
fb8b        int a,b,n,m,k;
cc1c        scanf("%d%d%d%d%d",&a,&n,&b,&m,&k);
5399        if(k == 0){
8acc            printf("Case␣%d:␣0\n",Case);
b333            continue;
95cf        }
0dac        n/=k;
a94f        m/=k;
0d4c        printf("Case␣%d:␣%lld\n",Case,work(n,m) - work(min(n,m),min(n,m))/2);
95cf    }
7021    return 0;
95cf}
```

8 Others

8.1 Header

```
427e// Created by calabash_boy
b54d#pragma GCC optimize(3)
302f#include <bits/stdc++.h>
421cusing namespace std;
426f#ifdef __LOCAL_DEBUG__
59a8# define _debug(fmt, ...) fprintf(stderr, "\033[91m[%s␣%3d]:␣" fmt "\n\033[0m",
    \
1a94  __func__,__LINE__, ##__VA_ARGS__)
a8cb#else
0c29# define _debug(...) (void(0))
1937#endif
d54b#define PB(x) push_back(x)
8f39#define rep(i,l,r) for (int i = l,_ = r;i< _;i++)
aa2e#define REP(i,l,r) for (int i=l,_=r;i<=_;i++)
7e99#define leave(x) do {cout<<#x<<endl;fflush(stdout);return 0;}while (0);
c33e#define untie do{ios::sync_with_stdio(false);cin.tie(nullptr);cout.tie(nullptr)
  ;}while (0)
aaca#define range(x) x.begin(),x.end()
5cadtypedef long long LL;
4085typedef long long ll;
76b3typedef vector<int> vi;
3a45typedef vector<ll> vl;
2bc8typedef long double db;
3688typedef pair<int,int> pii;
0d99typedef pair<ll,ll> pll;
a7c7const int inf = 0x3f3f3f3f;
a744const ll inf_ll = 0x3f3f3f3f3f3f3f3fLL;
5862/************* header *****************/
3117int main(){
7021    return 0;
95cf}
```