南京大学 ACM-ICPC 集训队
代码模版库

目录

1 String

1.1 Hash-1D

```
427e //
427e // Created by calabash_boy on 18-6-1.
427e // CF 1003F
427e //
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;
427e
5c83 int n;
4c95 int s[maxn];
87e7 /*
6f3b  * char* 1-bas
208b  * sum[i] = s[i]+s[i-1]*Seed+s[i-2]*Seed^2+...+s[1]*Seed^(i-1)
f2b5  */
d095 ULL Seed_Pool[]={911,146527,19260817,91815541};
c437 ULL Mod_Pool[]={29123,998244353,1000000009,4294967291ull};
b060 struct Hash_1D{
3e0c     ULL Seed,Mod;
2aae     ULL bas[maxn];
dd80     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){
7b7b         for (int i=1;i<sigma;i++){
871a             perm[i]=i;
95cf         }
e2fc         random_shuffle(perm+1,perm+1+n);
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];
643d         if (!id[a[i]]){
4897             id[a[i]] = ++idcnt;
95cf         }
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};
68f8             if (veid[x]==0){
```

```
c83f                veid[x] = ++vecnt;
95cf            }
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//
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。
427e//
302f#include<bits/stdc++.h>
427e//#define Debug(x) cerr<<#x<<" "<<x<<endl;
421cusing namespace std;
94a1const int maxn = 1e6+100;
427e
a239struct KMP{
51d9    int nxt[maxn];
57b7    int len;
    void clear(){                                                                    1126
        len =0;                                                                      61e2
        nxt[0] = nxt[1] =0;                                                          7f42
    }                                                                                95cf
    /* 1-bas */                                                                     c0bf
    /* 注意在ss结尾添加 '\0' */                                                     b115
    void init(char* ss){                                                             2e3f
        len = strlen(ss+1);                                                          64a4
        for (int i=2;i<=len;i++){                                                    ca76
            nxt[i] = nxt[i-1];                                                       362a
            while (nxt[i]&&ss[i]!=ss[nxt[i]+1]) nxt[i] = nxt[nxt[i]];                bbb0
            nxt[i]+=(ss[i]==ss[nxt[i]+1]);                                           da9f
        }                                                                            95cf
    }                                                                                95cf
    void debug(){                                                                    56dd
        for (int i=0;i<=len;i++){                                                    0d69
            printf("[debug] nxt[%d]=%d\n",i,nxt[i]);                                 3cb0
//            Debug(nxt[i]);                                                         427e
        }                                                                            95cf
    }                                                                                95cf
    /* 循环周期 形如 acaca 中 ac 是一个合法周期 */                                  243b
    vector<int> periodic(){                                                          d4e9
        vector<int> ret;                                                             995a
        int now = len;                                                               4a5d
        while (now){                                                                 3f78
            now = nxt[now];                                                          ebeb
            ret.push_back(len-now);                                                  9341
        }                                                                            95cf
        return ret;                                                                  ee0f
    }                                                                                95cf
    /* 循环节 形如 acac 中ac、acac是循环节，aca不是*/                               f525
    vector<int> periodic_loop(){                                                     1a85
        vector<int>ret ;                                                             995a
        for (int x :periodic()){                                                     d561
            if (len%x==0){                                                           284a
                ret.push_back(x);                                                    401f
            }                                                                        95cf
        }                                                                            95cf
        return ret;                                                                  ee0f
    }                                                                                95cf
    int min_periodic_loop(){                                                         5531
        return periodic_loop()[0];                                                   8b2c
    }                                                                                95cf
}kmper;                                                                              997f
```

```
0324 vector<string> s;
b647 vector<vector<int> > a;
9fa8 vector<vector<int> >maxVal;
f4d5 int cnt1[maxn],cnt2[maxn];
35b8 int n,m;
5f67 char S[maxn];
e6f2 pair<int,int> pq[maxn];int l,r;
3117 int main(){
a1c9 #ifdef ONLINE_JUDGE
7618     ios::sync_with_stdio(false);
498a     cin.tie(nullptr);
c16f     cout.tie(nullptr);
1937 #endif
9af0     cin>>n>>m;
9d25     s.resize(n+1);
035f     maxVal.resize(n+1);
6dbf     for (int i=1; i<=n;i++){
f9af         cin>>s[i];
95cf     }
246a     a.resize(n+1);
6dbf     for (int i=1;i<=n;i++){
4356         a[i].resize(m+1);
0901         maxVal[i].resize(m+1);
8e5f         for (int j=1;j<=m;j++){
0fb4             cin>>a[i][j];
95cf         }
95cf     }
fdb4     int p,q;
a24e     kmper.clear();
6dbf     for (int i=1;i<=n;i++){
8e5f         for (int j=1;j<=m;j++){
69f1             S[j] = s[i][j-1];
95cf         }
5239         S[m+1]='\0';
8dce         kmper.init(S);
1d4f         for (int x:kmper.periodic()){
3b83             cnt1[x]++;
95cf         }
95cf     }
8e5f     for (int j=1;j<=m;j++){
6dbf         for (int i=1;i<=n;i++){
3e08             S[i] = s[i][j-1];
95cf         }
80ba         S[n+1]='\0';
8dce         kmper.init(S);
1d4f         for (int x:kmper.periodic()){
e14e             cnt2[x]++;
95cf         }
95cf     }
b042     for (int i=maxn;i>=1;i--){
7f7a         if (cnt1[i]==n){
8dd2             q = i;
95cf         }
8918         if (cnt2[i]==m){
d29d             p=i;
95cf         }
95cf     }
6dbf     for (int i=1;i<=n;i++){
25ea         l = 0,r=0;
8e5f         for (int j=1;j<=m;j++){
872e             while (r>l&&pq[l].second<=j-q)l++;
26e9             while (r>l&&pq[r-1].first<=a[i][j])r--;
3497             pq[r++] = {a[i][j],j};
862b             if (j>=q){
1dcc                 maxVal[i][j-q+1] = pq[l].first;
95cf             }
95cf         }
95cf     }
54ad     int ans = 0x3f3f3f3f;
2f5d     for (int j=1;j<=m-q+1;j++){
edd7         l=r=0;
6dbf         for (int i=1;i<=n;i++){
be46             while (r>l&&pq[l].second<=i-p)l++;
bb56             while (r>l&&pq[r-1].first<=maxVal[i][j])r--;
c5e8             pq[r++] = {maxVal[i][j],i};
b6cf             if (i>=p){
3003                 ans = min(ans,pq[l].first);
95cf             }
427e
95cf         }
95cf     }
fc9a     cout<<1LL*(p+1)*(q+1)*ans<<endl;
7021     return 0;
95cf }
```

1.3 Manacher

```
427e //
427e // Created by calabash_boy on 18-9-14.
427e //
427e
302f #include<bits/stdc++.h>
421c using namespace std;
571f const int MAX = 2e5+10000;
04f3 char ch[MAX];
9ccd int lc[MAX];
5c83 int n;
df8b void Manacher(){
a461     lc[1]=1;  int k=1;
a5c5     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(){
6dbf     for (int i=1;i<=n;i++){
0d62         printf("lc[%d]=%d\n",i,lc[i]);
95cf     }
95cf }
3117 int main(){
80aa     scanf("%s",ch+1);
427e     //calc n must before call Manacher
4907     n = strlen(ch+1);
ad19     ch[n*2+1] = '#';
0c3f     for (int i=n;i>=1;i--){
6132         ch[i*2] = ch[i];
cbb0         ch[i*2-1] = '#';
95cf     }
fad8     n = n*2 +1;
b5bc     ch[0] = 'z'+1;
b839     ch[n+1] = '\0';
4f78     Manacher();
9946     debug();
7021     return 0;
95cf }
```

1.4 Suffix_Array

```
//                                                                  427e
// Created by calabash_boy on 18-7-3.                              427e
//                                                                  427e
#include<bits/stdc++.h>                                             302f
#define rank rkrk                                                   1abc
using namespace std;                                                421c
typedef long long ll;                                               4085
const int maxn=1e5+100;                                             52c1
char ch[maxn];                                                      6182
struct Node{                                                        80b8
    int val,index;                                                  314f
    Node(int val_,int index_):val(val_),index(index_){}             e831
    bool operator < (const Node b)const{                            d2bb
        if (val==b.val)return b.index<index;                        1ec4
        return b.val<val;                                           1e11
    }                                                               95cf
};                                                                  329b
priority_queue<Node>pq;                                             c124
namespace Suffix_Array{                                             5bf1
    int cntA[maxn],cntB[maxn],tsa[maxn],A[maxn],B[maxn];            6e4f
    int sa[maxn],rank[maxn],height[maxn];                           f3d8
    void GetSa(char *ch,int n){                                     7e17
        for(int i=0;i<maxn;i++)  cntA[i]=0;                         2ddf
        for(int i=1;i<=n;i++)   cntA[ch[i]]++;                      e86b
        for(int i=1;i<=maxn;i++) cntA[i]+=cntA[i-1];                edcc
        for(int i=n;i;i--)  sa[cntA[ch[i]]--]=i;                    94bb
        rank[sa[1]]=1;                                              c9f2
        for(int i=2;i<=n;i++){                                      a5c5
            rank[sa[i]]=rank[sa[i-1]];                              dc5c
            if(ch[sa[i]]!=ch[sa[i-1]])  rank[sa[i]]++;              459c
        }                                                           95cf
        for(int l=1;rank[sa[n]]<n;l<<=1){                           f62b
            for(int i=0;i<maxn;i++)  cntA[i]=0;                     2ddf
            for(int i=0;i<maxn;i++)  cntB[i]=0;                     db87
            for(int i=1;i<=n;i++){                                  6dbf
                cntA[A[i]=rank[i]]++;                               d9ab
                cntB[B[i]=(i+l<=n)?rank[i+l]:0]++;                  c846
            }                                                       95cf
            for(int i=1;i<maxn;i++)   cntB[i]+=cntB[i-1];           e54e
            for(int i=n;i;i--)  tsa[cntB[B[i]]--]=i;                1d70
            for(int i=1;i<maxn;i++)  cntA[i]+=cntA[i-1];            a49f
            for(int i=n;i;i--)  sa[cntA[A[tsa[i]]]--]=tsa[i];       b1ed
```

```
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    }
05e8    void GetHeight(char *ch,int n){
0b4d        GetSa(ch,n);
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        }
95cf    }
427e    //special
9d8d    int GetK(int k,int n){
3b0f        int ans=0;
c4cf        k--;
5399        if(k==0){
e8e9            for(int i=1;i<=n;++i)   ans=ans+(n-sa[i]+1-height[i]);
4206            return ans;
95cf        }
d805        while (!pq.empty())pq.pop();
a5c5        for (int i=2;i<=n;i++){
6821            while (!pq.empty()&&pq.top().index<i-k+1)pq.pop();
798c            pq.push(Node(height[i],i));
d772            if (i>k){
fddd                int top = pq.top().val;
4fae                int last = height[i-k];
5d00                ans +=max(0,top-last);
95cf            }
95cf        }
4206        return ans;
95cf    }
329b};
3117int main(){
9523    int T;
1fd9    scanf("%d",&T);
60ca    while(T--){
232a        int n,k;
c93a        scanf("%d",&k);
80aa        scanf("%s",ch+1);
            n=strlen(ch+1);                                                  4907
            Suffix_Array::GetHeight(ch,n);                                   9af4
            printf("%d\n",Suffix_Array::GetK(k,n)-Suffix_Array::GetK(k+1,n));  47ec
        }                                                                    95cf
        return 0;                                                            7021
}                                                                            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          427e
#include<bits/stdc++.h>                                                      302f
using namespace std;                                                         421c
const int maxn = 1e5+100;                                                    52c1
struct Aho_Corasick_Automaton{                                               6b3e
    //basic                                                                  427e
    int nxt[maxn*10][26],fail[maxn*10];                                      141b
    int root,tot;                                                            7a04
    //special                                                                427e
    int flag[maxn*10];                                                       8f42
    int len[maxn*10];                                                        d3a5
    void clear(){                                                            1126
        memset(nxt[0],0,sizeof nxt[0]);                                      21a1
        root = tot=0;                                                        0ae1
    }                                                                        95cf
    int newnode(){                                                           ee91
        tot++;                                                               71cf
        memset(nxt[tot],0,sizeof nxt[tot]);                                  87f4
        flag[tot] = len[tot]=0;                                              a231
        return tot;                                                          91fb
    }                                                                        95cf
    void insert(char *s ){                                                   9bb4
        int now = root;                                                      8f56
        while (*s){                                                          f205
            int id = *s-'a';                                                 e37a
            if(!nxt[now][id]){                                               0727
                nxt[now][id] = newnode();                                    9508
```

```
95cf            }
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';
0727            if(!nxt[now][id]){
9508                nxt[now][id] = newnode();
95cf            }
7134            len[nxt[now][id]] = len[now]+1;
6f00            now = nxt[now][id];
95cf        }
95cf    }
2114    void build(){
30ee        fail[root] = root;
aafa        queue<int>Q;
6568        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';
6f00        now = nxt[now][id];
c20a        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) {
3b0f    int ans =0;
8f56    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];
4d9b int m,n;
6393 int qid;
3117 int main(){
7618    ios::sync_with_stdio(false);
212b    cin.tie(0);
40ee    cout.tie(0);
9523    int T;
3f76    cin>>T;
60ca    while (T--){
7e53        acam.clear();
e1b6        cin>>n;
6dbf        for (int i=1;i<=n;i++){
879c            cin>>a[i];
e321            acam.insert(a[i]);
95cf        }
17ab        acam.build();
2eb3        cin>>m;
e052        for (int i=1;i<=m;i++){
0f8b            int x,y;
6a4f            qid++;
d480            cin>>x>>y;
```

```
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 //
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。
427e //
302f #include<bits/stdc++.h>
421c using namespace std;
40fb const int maxn = 25e4+100;
15df char s[maxn];
5c83 int n;
e8d4 int 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*/
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){
499b             add(*s-'a');
85be             s++;
95cf         }
95cf     }
681b     void add(int c){
a4cf         int p = last;
4428         int np = ++cnt;
8b9f         memset(nxt[cnt],0,sizeof nxt[cnt]);
3857         l[np] = l[p]+1;
544c         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]));
1033                 fa[nq] =fa[q];
ac00                 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]]++;
c35a     for (int i=1;i<=n;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 }
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);
3f76     sam.clear();
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 PAM

```
427e //
427e // Created by calabash_boy on 18-6-4.
427e // BZOJ 3676
427e // calc max(len(t)*cnt(t)) t为s回文子串，cnt(t)=t出现次数
427e //
302f #include<bits/stdc++.h>
421c using namespace std;
6428 const int maxn = 3e5+100;
466b struct Palindromic_AutoMaton{
427e     //basic
9f36     int s[maxn],now;
f801     int nxt[maxn][26],fail[maxn],l[maxn],last,tot;
427e     // extension
e216     int num[maxn];/*节点代表的所有回文串出现次数*/
1126     void clear(){
427e         //1节点：奇数长度root 0节点：偶数长度root
78a6         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){
5ae2             add(*ss-'a');
41eb             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;
26a1 long 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  }
15df  char s[maxn];
3117  int 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 Convex_Hull

```
427e  //
427e  // Created by calabash_boy on 18-9-14.
427e  //
427e
302f  #include<bits/stdc++.h>
421c  using namespace std;
5cad  typedef long long LL;
7144  const int maxn = 1005;
95b2  #define M_PI 3.1415926535
b400  struct Node{int x,y;};
f306  int st[maxn],top; Node a[maxn];
6e48  int rk[maxn];int n,T,l;
4b6d  LL cross(const Node &a,const Node &b,const Node &c){
9970      return 1LL*(b.x-a.x)*(c.y-a.y)-1LL*(c.x-a.x)*(b.y-a.y);
95cf  }
2d56  LL cross(int x,int y,int z){return cross(a[x],a[y],a[z]);}
f7d7  double dis(const Node &a,const Node &b){
a055      return sqrt(1.0*(a.x-b.x)*(a.x-b.x)+1.0*(a.y-b.y)*(a.y-b.y));
95cf  }
f88e  bool cmp(int x,int y){
9692      LL m = cross(a[rk[0]],a[x],a[y]);
3f57      if (m>0)return 1;
ed4d      else if (m==0&&dis(a[rk[0]],a[x])<=dis(a[rk[0]],a[y]))return 1;
426e      else return 0;
95cf  }
9627  void solve(){
5256      scanf("%d%d",&n,&l);
1294      for (int i=0;i<n;i++){
1387          scanf("%d%d",&a[i].x,&a[i].y);
f9d0          rk[i]=i;
95cf      }
324a      for (int i=1;i<n;i++){
7d84          if (a[rk[i]].y<a[rk[0]].y||a[rk[i]].y==a[rk[0]].y&&a[rk[i]].x<a[rk[0]].x
                )swap(rk[i],rk[0]);
95cf      }
fd2f      sort(rk+1,rk+n,cmp);top=2;
828b      st[0]=rk[0];st[1]=rk[1];
4585      for (int i=2;i<n;i++){
2401          while (cross(st[top-2],st[top-1],rk[i])<0)top--;
3986          st[top++] =rk[i];
95cf      }
753f      double ans =0;
e1f3      for (int i=1;i<top;i++){ans+=dis(a[st[i]],a[st[i-1]]);}
fe12      ans+=dis(a[st[0]],a[st[top-1]]);
e10a      ans+=2*M_PI*l;
adb0      printf("%.0lf\n",ans);
95cf  }
3117  int main(){
1fd9      scanf("%d",&T);
60ca      while (T--){
ccd1          solve();
408c          if (T!=0)printf("\n");
95cf      }
7021      return 0;
95cf  }
```

3.2 Max_Flow

```
427e  //
427e  // Created by calabash_boy on 18-9-14.
427e  //
302f  #include<bits/stdc++.h>
421c  using namespace std;
00ad  const int maxn = 205;
08a4  const int INF = 0x3f3f3f3f;
cdb6  int first[maxn],nxt[maxn*2],des[maxn*2],c[maxn*2],tot;
1153  int dep[maxn];int m,n,ss,tt;
5d53  void init(){
8eac      memset(first,-1,sizeof first);
```

```
    tot =-1;                                                                    ee65
}                                                                               95cf
inline void addEdge(int u,int v,int w){                                         4a69
    tot++;                                                                      71cf
    des[tot] = v;c[tot] =w;                                                     73e4
    nxt[tot] = first[u];first[u] = tot;                                         6570
}                                                                               95cf
void input(){                                                                   0e91
    for (int i=0;i<m;i++){                                                      356f
        int u,v,w;                                                              3676
        scanf("%d%d%d",&u,&v,&w);                                               95a1
        addEdge(u,v,w);addEdge(v,u,0);                                          16fe
    }                                                                           95cf
}                                                                               95cf
bool bfs(){                                                                     1836
    memset(dep,-1,sizeof dep);                                                  d568
    dep[ss] =0;                                                                 0881
    queue<int> Q;Q.push(ss);                                                    fc6b
    while (!Q.empty()){                                                         11e5
        int q = Q.front();Q.pop();                                              d7b1
        for (int t = first[q];t!=-1;t= nxt[t]){                                 9c72
            int v = des[t],cx = c[t];                                           b7bb
            if (dep[v]==-1&&cx){                                                c804
                dep[v] = dep[q]+1;                                              31e8
                Q.push(v);                                                      78e5
            }                                                                   95cf
        }                                                                       95cf
    }                                                                           95cf
    return dep[tt]!=-1;                                                         45fe
}                                                                               95cf
int dfs(int node,int now){                                                      c29e
    if (node==tt)return now;                                                    0031
    int res =0;                                                                 5839
    for (int t = first[node];t!=-1&&res<now;t=nxt[t]){                          1e7e
        int v = des[t],cx = c[t];                                               b7bb
        if (dep[v]==dep[node]+1&&cx){                                           da1a
            int x = min(cx,now-res);                                            223c
            x = dfs(v,x);                                                       6c2e
            res +=x;                                                            68f7
            c[t] -= x;c[t^1]+=x;                                                2a05
        }                                                                       95cf
    }                                                                           95cf
    if (!res) dep[node] = -2;                                                   7399
    return res;                                                                 244d
}                                                                               95cf
void solve(){                                                                   9627
    int res =0,del =0;                                                          c6c0
    ss =1;tt =n;                                                                0f48
    while (bfs()){                                                              ed58
        while (del = dfs(ss,INF)){res +=del;}                                   67df
    }                                                                           95cf
    cout<<res<<endl;                                                            b830
}                                                                               95cf
int main(){                                                                     3117
    while (scanf("%d%d",&m,&n)!=EOF){                                           1fb7
        init();                                                                 07e2
        input();                                                                2a5c
        solve();                                                                ccd1
    }                                                                           95cf
    return 0;                                                                   7021
}                                                                               95cf
```

3.3 Min_Cost_Max_Flow

```
//                                                                              427e
// Created by calabash_boy on 18-9-14.                                         427e
//                                                                              427e
#include<cstdio>                                                                59b9
#include<iostream>                                                              e0a5
#include<cstring>                                                               ef2f
#include<algorithm>                                                             54ff
#include<queue>                                                                 acb9
using namespace std;                                                            421c
const int maxn = 2000+50;                                                       90ff
const int maxm = 20000+50;                                                      4ba7
const int INF = 0x3f3f3f3f;                                                     08a4
int m,n;                                                                        4d9b
int first[maxn],from[maxm*2],des[maxm*2],nxt[maxm*2],cost[maxm*2],flow[maxm*2],  4b98
  tot;
int dis[maxn],pre[maxn];                                                        ed91
bool in[maxn];int ss,tt;                                                        e132
inline void addE(int x,int y,int f,int c){                                      abbb
    tot++;                                                                      71cf
    from[tot] =x;des[tot] =y;                                                   575f
    flow[tot] =f;cost[tot] =c;                                                  4b45
    nxt[tot] = first[x];first[x] = tot;                                         6d84
}                                                                               95cf
```

```
f1f8 inline void addEdge(int x,int y,int f,int c){
8dad     addE(x,y,f,c);addE(y,x,0,-c);
95cf }
0e91 void input(){
ac98     scanf("%d%d",&n,&m);
ee65     tot =-1;
8eac     memset(first,-1,sizeof first);
356f     for (int i=0;i<m;i++){
a083         int u,v,c;
1493         scanf("%d%d%d",&u,&v,&c);
252c         addEdge(u,v,1,c);addEdge(v,u,1,c);
95cf     }
0fbc     addEdge(0,1,2,0);
95cf }
3c52 bool spfa(){
f25d     memset(in,0,sizeof in);
9ca1     memset(dis,INF,sizeof dis);
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();
f2f8         Q.pop();
66e0         in[q] = 0;
9c72         for (int t = first[q];t!=-1;t = nxt[t]){
e8e0             int v = des[t];
c471             int len = cost[t];
0021             int 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     }
16b4     return pre[tt]!=-1;
95cf }
9627 void solve(){
ba51     ss =0;tt=n;
eb96     int totflow =0,totcost =0,nowflow =0,nowcost =0;
22dc     while (spfa()){
4b98         nowcost =0;
4aff         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     }
ef8d     cout<<totcost<<endl;
95cf }
3117 int main(){
2a5c     input();
ccd1     solve();
7021     return 0;
95cf }
```

3.4 LCA

```
427e //
427e // Created by calabash_boy on 18-7-7.
427e //
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){
71cf     tot++;
c54b     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);
d315         addEdge(x,y);
ba13         addEdge(y,x);
95cf     }
73b1     Multiply_LCA::dfs(s,0);
3f3a     while (m--){
0f8b         int x,y;
a9b3         scanf("%d%d",&x,&y);
d93e         printf("%d\n",Multiply_LCA::lca(x,y));
95cf     }
7021     return 0;
}                                                                    95cf
```

3.5 DSU_On_Tree(General)

```
//                                                                   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                          427e
//                                                                   427e
#include <bits/stdc++.h>                                             302f
using namespace std;                                                 421c
typedef long long ll;                                                4085
typedef pair<int,int> pii;                                           3688
#define rep(i,l,r) for (ll i = l,_ = r;i< _;i++)                     31ec
#define REP(i,l,r) for (ll i=l,_=r;i<=_;i++)                         5879
/************* header ******************/                           5862
const int maxn = 5e5+100;                                            6f64
int n,tot,first[maxn],des[maxn],nxt[maxn],m;                         2ff9
vector<pii> Q[maxn];                                                 28d5
int cnt[maxn][26],Cnt[maxn];                                         f96d
int sz[maxn],dep[maxn],wson[maxn];                                   bbe3
bool ans[maxn];                                                      cd1e
char s[maxn];                                                        15df
bool big[maxn];                                                      f6e9
inline void addEdge(int x,int y){                                    453e
    tot++;                                                           71cf
    des[tot] = y;                                                    c54b
    nxt[tot] = first[x];                                             465b
    first[x] = tot;                                                  86fa
}                                                                    95cf
void get_sz(int node,int depth){                                     0d39
    dep[node] = depth;                                               93f9
    sz[node] = 1;                                                    889d
    for (int t = first[node];t;t=nxt[t]){                            e83e
        int v = des[t];                                              e8e0
        get_sz(v,depth+1);                                           a0d5
        sz[node] += sz[v];                                           47d5
        if (sz[v] > sz[wson[node]]){                                 acb3
            wson[node] = v;                                          44c0
        }                                                            95cf
    }                                                                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      }
2578      get_sz(1,1);
99d6      dfs(1,0);
a826      rep(i,0,m){
3db8          printf("%s\n",ans[i]?"Yes":"No");
95cf      }
7021      return 0;
95cf  }
```

3.6 DSU_On_Tree(Rough)

```
427e  //
427e  // Created by calabash_boy on 18-10-7.
427e  //
523c  /* CF 600E
7a5e   * dsu on tree
eb58   * calc the sum of color_id whose occurencing time is biggest in every subtree
c4c5   * dsu: nlogn map:logn total: nlog^2n
f2b5   */
302f  #include <bits/stdc++.h>
421c  using namespace std;
4085  typedef long long ll;
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++)
c33e  #define untie do{ios::sync_with_stdio(false);cin.tie(nullptr);cout.tie(nullptr)
        ;}while (0)
5862  /************* header *****************/
427e
52c1  const int maxn = 1e5+100;
19dc  int a[maxn],first[maxn],des[maxn*2],nxt[maxn*2],tot;
5c83  int n;
301f  map<int,int> *cnt[maxn];
e652  ll ans[maxn];
94a8  int mx[maxn];
e67c  int sz[maxn],wson[maxn];
453e  inline void addEdge(int x,int y){
71cf      tot ++;
c54b      des[tot] = y;
465b      nxt[tot] = first[x];
86fa      first[x] = tot;
95cf  }
da08  inline void relax(int v,int t,int cnt){
a29f      if (cnt>mx[v]){
eef8          mx[v] = cnt;
db44          ans[v] = t;
22ce      }else if (cnt == mx[v]){
a8e8          ans[v] +=t;
95cf      }
```

```
95cf }
dd7c void dfs(int node,int father){
889d     sz[node] = 1;
e83e     for (int t = first[node];t;t=nxt[t]){
e8e0         int v = des[t];
ca31         if (v == father)continue;
1f8e         dfs(v,node);
47d5         sz[node] += sz[v];
acb3         if (sz[v] > sz[wson[node]]){
44c0             wson[node] = v;
95cf         }
95cf     }
d010     if (wson[node]){
9088         cnt[node] = cnt[wson[node]];
4ea1         ans[node] = ans[wson[node]];
c897         mx[node] = mx[wson[node]];
8e2e     }else{
bbdb         cnt[node] = new map<int,int>();
95cf     }
2bc7     (*cnt[node])[a[node]]++;
b69a     relax(node,a[node],(*cnt[node])[a[node]]);
e83e     for (int t = first[node];t;t=nxt[t]){
e8e0         int v = des[t];
423c         if (v == father || v == wson[node])continue;
7ce9         for (auto pair : *cnt[v]){
2e74             (*cnt[node])[pair.first] += pair.second;
ce15             relax(node,pair.first,(*cnt[node])[pair.first]);
95cf         }
95cf     }
95cf }
3117 int main(){
79d8     untie;
e1b6     cin>>n;
8117     REP(i,1,n)cin>>a[i];
656a     rep(i,1,n){
0f8b         int x,y;
d480         cin>>x>>y;
d315         addEdge(x,y);
ba13         addEdge(y,x);
95cf     }
99d6     dfs(1,0);
fce9     REP(i,1,n)cout<<ans[i]<<"␣";
3251     cout<<endl;
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))                                               427e
#include<bits/stdc++.h>                                             302f
using namespace std;                                                421c
const int MAX = 1e6+100;                                            ed66
int bas[35];                                                        e0df
int n,Cas;                                                          1468
const int INF = 2147483645;                                         92ad
struct Trie{                                                        a281
    int nxt[MAX<<2][2];  int l[MAX<<2];                             30cd
    int cnt;  int ansl,ansr,ansv;                                   c92e
    void init(){                                                    5d53
        cnt =0;                                                     8766
        memset(nxt[0],0,sizeof (nxt[0]));                           16d8
        memset(l,0x3f3f3f3f,sizeof(l));                             aa76
        ansv = 0;                                                   840a
    }                                                               95cf
    int create(){                                                   b87c
        cnt++;                                                      6fb3
        memset(nxt[cnt],0,sizeof (nxt[cnt]));                       3b79
        return cnt;                                                 6808
    }                                                               95cf
    void insert(int id,int x){                                      d5dd
        int y = 0;                                                  875c
        for (int i=30;i>=0;i--){                                    7ecf
            int t = x&bas[i];                                       0c9f
            t>>=i;                                                  2e46
            if (!nxt[y][t]){                                        a5f0
                nxt[y][t] = create();                               eb8b
            }                                                       95cf
            y = nxt[y][t];                                          f056
        }                                                           95cf
        l[y] = min(l[y],id);                                        a4a7
    }                                                               95cf
    void query(int id,int x){                                       1a97
```

```
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;
427e
3117 int main(){
bf6d     bas[0] = 1;
dc7e     for (int i1=1;i1<=30;i1++){
abeb         bas[i1] = bas[i1-1]<<1;
95cf     }
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。                                                      427e
//                                                                          427e
                                                                            427e
#include<bits/stdc++.h>                                                     302f
using namespace std;                                                        421c
#define OPENSTACK                                                           1585
                                                                            427e
const int maxn = 1e6+100;                                                   94a1
const int mod = 1e9+7;                                                      5d33
typedef long long LL;                                                       5cad
int stk[maxn],top;                                                          f706
int l[maxn],r[maxn],rt;                                                     4927
int n;                                                                      5c83
pair<int,int> a[maxn];                                                      62bd
LL inv[maxn];                                                               7c76
LL fac[maxn];                                                               ec8f
LL inv_fac[maxn];                                                           e6de
int sz[maxn];                                                               590c
bool vis[maxn];                                                             dbd8
/* l 左儿子 r 右儿子 rt根*/                                                 ea2f
void build(){                                                               2114
    top=0;                                                                  3e5f
    for (int i=1;i<=n;i++) l[i]=r[i]=vis[i] =0;                             4c1f
    for (int i=1;i<=n;i++){                                                 6dbf
        int k = top;                                                        8077
        while (k&&a[i]<a[stk[k-1]])k--;                                     14fa
        if (k) r[stk[k-1]] = i;                                             004e
        if (k<top) l[i] = stk[k];                                           90d1
        stk[k++] =i;                                                        18d7
        top = k;                                                            ad1c
    }                                                                       95cf
    for (int i=1;i<=n;i++) vis[l[i]] = vis[r[i]] =1;                        791b
    for (int i=1;i<=n;i++){                                                 6dbf
        if (!vis[i]){                                                       794b
            rt = i;                                                         cf39
            break;                                                          6173
        }                                                                   95cf
    }                                                                       95cf
}                                                                           95cf
LL power(LL x,LL y){                                                        a89a
```

```
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){
50c0      sz[u]=1;
f67f      int ans =1;
fe92      if (l[u])ans=1LL*ans*dfs(l[u])%mod;
429f      if (r[u])ans = 1LL*ans*dfs(r[u])%mod;
2c7a      sz[u]+=sz[l[u]]+sz[r[u]];
b778      return 1LL*ans*C(sz[u]-1,sz[l[u]])%mod;
95cf  }
6e6d  void Main(){
acce      inv[1]=fac[1]=fac[0]=1;
3295      for (int i=2;i<maxn;i++)fac[i] = fac[i-1]*i%mod,inv[i] = inv[mod%i]*(mod-mod
            /i)%mod;
5f9e      inv_fac[maxn-1] = power(fac[maxn-1],mod-2);
c2aa      for (int i=maxn-2;i>=0;i--){
4cf8          inv_fac[i] = inv_fac[i+1]*(i+1)%mod;
95cf      }
9523      int T;
1fd9      scanf("%d",&T);
60ca      while (T--) {
cd91          scanf("%d", &n);
6dbf          for (int i = 1; i <= n; i++) {
3c9e              int x;
ea4e              scanf("%d", &x);
d6d4              a[i] = {-x, i};
95cf          }
7068          build();
b475          printf("%d\n", inv[2] * n % mod * power(fac[n], mod - 2) % mod * dfs(rt)
                 % mod);
95cf      }
95cf  }
3117  int main(){
4b95  #ifdef OPENSTACK
90c5      int size = 70 << 20; // 256MB
      char *p = (char*)malloc(size) + size;                                  9efa
#if (defined _WIN64) or (defined __unix)                                     8c82
      __asm__("movq␣%0,␣%%rsp\n" :: "r"(p));                                 665b
#else                                                                        a8cb
      __asm__("movl␣%0,␣%%esp\n" :: "r"(p));                                 355e
#endif                                                                       1937
#endif                                                                       1937

      Main();                                                                362c
#ifdef OPENSTACK                                                             4b95
      exit(0);                                                               a398
#else                                                                        a8cb
      return 0;                                                              7021
#endif                                                                       1937

}                                                                            95cf
```

4.3 Chairman_Tree

```
//                                                                           427e
// Created by calabash_boy on 18-7-7.                                       427e
// query_kth_element                                                         427e
#include<bits/stdc++.h>                                                      302f
using namespace std;                                                         421c
const int maxn=1e5+100;                                                      52c1
int a[maxn];int rk[maxn];int pos[maxn];                                      b425
int root[maxn];int cnt,m,n,T;                                                15ac
struct Chairman_Tree{                                                        6207
    struct Node{int L,R,val;}tree[maxn*500];                                 108d
    void init(){                                                             5d53
        memset(root,0,sizeof root);                                          a4f5
        cnt =0;                                                              8766
    }                                                                        95cf
    /* 建T0空树 */                                                           94cf
    int buildT0(int l, int r){                                               cf84
        int k = cnt++;                                                       64f2
        tree[k].val =0;                                                      e9d1
        if (l==r) return k;                                                  eb40
        int mid = l+r >>1;                                                   b8b7
        tree[k].L = buildT0(l, mid);tree[k].R = buildT0(mid + 1, r);         1e97
        return k;                                                            e27b
    }                                                                        95cf
    /* 上一个版本节点P，【ppos】+=del 返回新版本节点*/                        e965
```

```
3a6b     int update (int P,int l,int r,int ppos,int del){
64f2         int k = cnt++;
1e22         tree[k].val = tree[P].val +del;
eb40         if (l==r) return k;
b8b7         int mid = l+r >>1;
4af7         if (ppos<=mid){
59bb             tree[k].L = update(tree[P].L,l,mid,ppos,del);
1cb7             tree[k].R = tree[P].R;
8e2e         }else{
a8f5             tree[k].L = tree[P].L;
d096             tree[k].R = update(tree[P].R,mid+1,r,ppos,del);
95cf         }
e27b         return k;
95cf     }
4798     int query_kth(int lt,int rt,int l,int r,int k){
9e61         if (l==r) return a[rk[l]];
b8b7         int mid = l+r >>1;
9988         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--){
8f36             int l,r,k;
edb0             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 //
427e // Created by calabash_boy on 18-10-6.
427e //
427e
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;
4922 }hotel[maxn],kdtree[maxn];
2ece double var[demension];
8003 int split [maxn];int cmpDem;
5cdc bool cmp(const Hotel &a,const Hotel &b){
b5cd     return a.pos[cmpDem]<b.pos[cmpDem];
95cf }
d5af void build (int l,int r){
2625     if (l>=r)return;
b8b7     int mid = l+r >>1;
8037     for (int i=0;i<demension;i++){
4655         double ave =0;
a0d3         for (int j=l;j<=r;j++){
70b6             ave+=hotel[j].pos[i];
95cf         }
b1eb         ave/=(r-l+1);var[i] =0;
a0d3         for (int j=l;j<=r;j++){
27fe             var[i]+=pow(hotel[j].pos[i]-ave,2);
95cf         }
6e08         var[i]/=(r-l+1);
95cf     }
3909     split[mid] =-1;double maxVar=-1;
8037     for (int i=0;i<demension;i++){
```

```
d704         if (var[i]>maxVar){
3bdc             maxVar = var[i];
9c04             split[mid] =i;
95cf         }
95cf     }
82fa     cmpDem = split[mid];
d815     nth_element(hotel+l,hotel+mid,hotel+r+1,cmp);
7bac     build (l,mid-1);build (mid+1,r);
95cf }
b10a int ansIndex;
5721 LL ansDis;
c274 void query(int l,int r,const Hotel& x){
8b8a     if (l>r)return ;
c410     int mid = l+r >>1;LL dis =0;
8037     for (int i=0;i<demension;i++){
3cc8         dis +=1LL*(x.pos[i]-hotel[mid].pos[i])*(x.pos[i]-hotel[mid].pos[i]);
95cf     }
9fff     if (hotel[mid].c<=x.c){
6bed         if (ansDis == dis && hotel[mid].id<hotel[ansIndex].id){
f191             ansIndex = mid;
f598         }else if (dis<ansDis){
de61             ansDis = dis;
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 //
427e // Created by calabash_boy on 18-9-14.
427e //
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;
2e15 char opt[5];
4d9b 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    }
    return 0;                                                    7021
}                                                                95cf
```

4.6 AFL(Cactus)

```
//                                                               427e
// Created by calabash_boy on 18-9-14.                          427e
//                                                               427e
// circle-square-tree Maximum independent set                   427e
#include<bits/stdc++.h>                                          302f
using namespace std;                                             421c
const int maxn = 1e5+100;                                        52c1
vector<int> E1[maxn],ET[maxn];                                   9010
int m,n,N;                                                       c7f9
int len[maxn],dfn[maxn],dfs_clock;                               d746
bool inCircle[maxn];                                             e6da
int fa[maxn];                                                    33ef
int dp[maxn][2];                                                 e3d4
int dp2[maxn][2];                                                4ab4
inline void addEdge1(int x,int y){                               e227
    E1[x].push_back(y);                                          f4a7
}                                                                95cf
inline void addEdgeT(int x,int y){                               2a27
    ET[x].push_back(y);                                          de38
}                                                                95cf
void input(){                                                    0e91
    cin>>n>>m;                                                   9af0
    N =n;                                                        7839
    for (int i=0;i<m;i++){                                       356f
        int u,v;                                                 54f1
        cin>>u>>v;                                               a02c
        addEdge1(u,v);                                           1a88
        addEdge1(v,u);                                           d47c
    }                                                            95cf
}                                                                95cf
void tarjan(int u){                                              74b1
    dfn[u] = ++dfs_clock;                                        f5c7
    for (int i=0;i<E1[u].size();i++){                            1958
        int v = E1[u][i];                                        1654
        if (v==fa[u])continue;                                   8e32
        if (!dfn[v]){                                            3c64
            fa[v] = u;                                           bac1
            tarjan(v);                                           67bb
```

```
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){
6684        dp[u][0]=0;
            dp[u][1]=1;                                                    14e3
            if (u>N)dp[u][0]=0;                                            16e7
            for (int i=0;i<ET[u].size();i++){                              5ee5
                int v = ET[u][i];                                          f37f
                dfs(v);                                                    5f3c
                if (u<=N){                                                 2900
                    dp[u][0]+=max(dp[v][1],dp[v][0]);                      edd9
                    dp[u][1]+=dp[v][0];                                    2a1b
                }                                                          95cf
            }                                                              95cf
            if (u>N){                                                      c9f5
                work(u);                                                   88cd
            }                                                              95cf
        }                                                                  95cf
        int main(){                                                        3117
            input();                                                       2a5c
            tarjan(1);                                                     951d
            dfs(1);                                                        dcdd
            cout<<max(dp[1][0],dp[1][1])<<endl;                            09a1
            return 0;                                                      7021
        }                                                                  95cf
```

4.7 Segment_Tree(Dynamic_Memory).cpp

```
//                                                                         427e
// Created by calabash_boy on 18-10-1.                                    427e
//                                                                         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                                                          427e
#include <bits/stdc++.h>                                                   302f
using namespace std;                                                       421c
const int maxn = 1e5+100;                                                  52c1
typedef long long ll;                                                      4085
struct Node{                                                               80b8
    int L,R,val;                                                           e7f7
}tree[maxn*200];                                                           7545
int cnt;                                                                   9f58
struct Segment_Tree{                                                       9c29
    int root = 0;                                                          e7b0
    int newnode(){                                                         ee91
```

```
06cb         ++cnt;
6598         tree[cnt].val = tree[cnt].L = tree[cnt].R = 0;
6808         return cnt;
95cf     }
6424     Segment_Tree(){
aa59         root = newnode();
95cf     }
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++){
d568         int x,r,q;
9fd9         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++){
d568         int x,r,q;
0c59         tie(x,r,q) = a[i];
d09f         int L = id[x-r];
6ce9         int 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 }
```

5 Graph

5.1 Tarjan(BCC_Edge)

```
427e // 
427e // Created by calabash_boy on 18-10-10.
427e //
302f #include<bits/stdc++.h>
421c using namespace std;
```

```
52c1 const int maxn = 1e5+100;
5b3f int first[maxn],nxt[maxn*2],from[maxn*2],des[maxn*2],isBrige[maxn*2],tot;
ff12 int dfn[maxn],low[maxn],dfs_clock;
8c69 int cnt_e[maxn],cnt_n[maxn];int bcc_cnt;
e093 bool ok[maxn];vector <int> ans;int m,n;
453e inline void addEdge(int x,int y){
71cf     tot++;
56e8     des[tot] =y;from[tot] =x;
6d84     nxt[tot] = first[x];first[x] = tot;
95cf }
0e91 void input(){
9af0     cin>>n>>m;
356f     for (int i=0;i<m;i++){
54f1         int u,v;
e9a7         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]){
071c         int v = des[t];if (v==fa)continue;
3c64         if (!dfn[v]){
e2f7             dfs(v,u);
7078             low[u] = min(low[v],low[u]);
f611             if (dfn[u]<low[v]){
4639                 isBrige[t] = true;
b158                 if (t&1){isBrige[t+1] = true;}
6c47                 else{isBrige[t-1] = true;}
95cf             }
e138         }else if (dfn[v]<dfn[u]){low[u] = min(low[u],dfn[v]);}
95cf     }
95cf }
e992 void blood_fill(int x){
ec01     dfn[x] = bcc_cnt;
4bb0     for (int t = first[x];t;t=nxt[t]){
9516         if (isBrige[t])continue;
e8e0         int v = des[t];
7127         if (!dfn[v]){blood_fill(v);}
95cf     }
95cf }
fd4b void check(){
a599     for (int i=1;i<=n;i++){cnt_n[dfn[i]]++;}
a7c6     for (int i=1;i<=tot;i++){
         if (isBrige[i]) continue;                                  7701
         cnt_e[dfn[des[i]]]++;                                      5746
     }                                                              95cf
     for (int i=1;i<=bcc_cnt;i++){                                  41ce
         if (cnt_n[i]*2==cnt_e[i]){ok[i]=1;}                        e64d
     }                                                              95cf
}                                                                   95cf
void output(){                                                      d880
     for (int i=1;i<=tot;i+=2){                                     8d09
         if (isBrige[i])continue;                                   7701
         if (ok[dfn[des[i]]])ans.push_back((i+1)/2);                c2ef
     }                                                              95cf
     sort(ans.begin(),ans.end());                                   e139
     cout<<ans.size()<<endl;                                        c4d5
     for (int i=0;i<ans.size();i++){printf("%d ",ans[i]);}          263e
}                                                                   95cf
void solve(){                                                       9627
     for (int i=1;i<=n;i++){if (!dfn[i])dfs(i,-1);}                 c2a0
     memset(dfn,0,sizeof dfn);                                      cbec
     for (int i=1;i<=n;i++){                                        6dbf
         if (!dfn[i]){                                              aa35
             bcc_cnt++;                                             03f5
             blood_fill(i);                                         3b53
         }                                                          95cf
     }                                                              95cf
     check();output();                                              92ea
}                                                                   95cf
int main(){                                                         3117
     input();                                                       2a5c
     solve();                                                       ccd1
     return 0;                                                      7021
}                                                                   95cf
```

5.2 Tarjan(BCC_Point)

```
//                                                                  427e
// Created by calabash_boy on 18-10-10.                            427e
//                                                                  427e
#include<bits/stdc++.h>                                             302f
using namespace std;                                                421c
const int maxn = 1e5+100;                                           52c1
int first[maxn],des[maxn*2],nxt[maxn*2],tot;                        58a9
int bcc_cnt,cnt_n[maxn],cnt_e[maxn],bcc_no[maxn];                   09ab
```

```
int dfn[maxn],low[maxn],dfs_clock;
int st[maxn*2],top;bool ok[maxn];
vector<int> ans;vector<int> temp;
int m,n;
inline void addEdge(int x,int y){
    tot++;des[tot] = y;
    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]){
            st[top++] = t;dfs(v,u);
            low[u] = min(low[u],low[v]);
            if (low[v]>=dfn[u]){
                bcc_cnt++;ok[bcc_cnt] = true;
                temp.clear();
                while (true){
                    int tt = st[--top];
                    temp.push_back((tt+1)/2);
                    if (bcc_no[des[tt]]!=bcc_cnt){
                        bcc_no[des[tt]] = bcc_cnt;
                        cnt_n[bcc_cnt]++;
                    }else{
                        ok[bcc_cnt] = false;
                    }
                    cnt_e[bcc_cnt]++;
                    if (tt==t){
                        break;
                    }
                }
                if (ok[bcc_cnt]&&temp.size()>1){
                    for (int i=0;i<temp.size();i++){
                        ans.push_back(temp[i]);
                    }
                }
            }
        }else if (dfn[v]<dfn[u]){
            st[top++] = t;
            low[u] = min(low[u],dfn[v]);
        }
    }
}
void solve(){
    for (int i=1;i<=n;i++){if (!dfn[i])dfs(i,-1);}
    sort(ans.begin(),ans.end());
    cout<<ans.size()<<endl;
    for (int i=0;i<ans.size();i++){printf("%d ",ans[i]);}
}
int main(){
    input();
    solve();
    return 0;
}
```

5.3 Tarjan(SCC)

```
#include<bits/stdc++.h>
using namespace std;
const int maxn = 1e5+100;
int m,n,h;int t[maxn];
int first[maxn*2],nxt[maxn*2],des[maxn*2],tot;
int dfn[maxn],low[maxn],dft;bool d[maxn];
int flag[maxn],cnt[maxn],scc;stack<int> stk;
inline void add(int x,int y){
    tot++;des[tot] =y;
    nxt[tot] = first[x];first[x] =tot;
}
void tar(int node){
    dfn[node] = low[node] = ++dft;
    stk.push(node);
    for (int t = first[node];t;t=nxt[t]){
        int v = des[t];
        if (!dfn[v])tar(v);
        low[node] = min(low[node],low[v]);
    }
    if (dfn[node]==low[node]){
```

```
38ac            scc++;
1026            while (true){
6947                int temp = stk.top();
80c2                flag[temp]=scc;
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++){
da47            int u1,u2;
d0e6            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;
a099                else{d[flag[i]]++;}
95cf            }
95cf        }
61a1        cnt[0] =n+1;int ans = 0;
5176        for (int i=1;i<=scc;i++){
83aa            if (d[i]==0&&cnt[i]<cnt[ans]){ans = i;}
95cf        }
31ae        cout<<cnt[ans]<<endl;
6dbf        for (int i=1;i<=n;i++){
e341            if (flag[i]==ans){cout<<i<<"␣";}
95cf        }
3251        cout<<endl;
7021        return 0;
95cf    }
```

6 Graph/Tree

6.1 Point-Divide&Conquer

```
427e    //
```

```
427e    // Created by calabash_boy on 18-10-6.
427e    //
427e    //求树上长度小于等于k的有向路径数
1915    #include<stdio.h>
54ff    #include<algorithm>
ef2f    #include<cstring>
421c    using namespace std;
bbaa    const int MAX = 1e4+100;
08a4    const int INF = 0x3f3f3f3f;
0b89    int first [MAX*2];  int des[MAX*2];
3efe    int len[MAX*2];  int nxt[MAX*2];
956f    int n,k,tot;  int a[MAX];  int sum[MAX];
ecb3    int dp[MAX];  int dis[MAX];  int num,ans;
aa8d    bool vis[MAX];  int Sum,Min,Minid;
5d53    void init(){
57d5        memset(first,0,sizeof first);
7ae1        tot =0;  ans =0;
87fb        memset(vis,0,sizeof vis);
95cf    }
ce82    inline void add(int x,int y,int z){
71cf        tot++;
3615        des[tot]= y;  len[tot] =z;
6d84        nxt[tot] = first[x];  first[x] = tot;
95cf    }
0e91    void input(){
324a        for (int i=1;i<n;i++){
3676            int u,v,w;
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 Tree_Chain_Division

```
427e  //
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;
7b14  int first[maxn*2];int nxt[maxn*2];int des[maxn*2];
0d93  int tpos[maxn];int dep[maxn];int top[maxn];
d6bf  int fa[maxn]; int wson[maxn];  int sz[maxn];
4ea4  int n,q,m,Root,tot=0,cnt=0;  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){
3dd2          for (int i=l;i<=r;i++){
325f              add(i,1);
95cf          }
95cf      }
6142      void add(int x,int val){
dc9a          while (x<=maxn){
865e              sm[x]+=val;
e6d9              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;
427e
f9d3    inline void addEdge(int _u, int _v){
26b9        des[++tot] = _v;
a66a        nxt[tot] = first[_u];
593b        first[_u] = tot;
95cf    }
11f1    namespace Tree_Chain_Division{
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    }
427e
427e
3117    int main(){
cd91        scanf("%d",&n);
324a        for (int i=1;i<n;i++){
17be            int u,v;  scanf("%d%d",&u,&v);
ad4e            addEdge(u, v);addEdge(v, u);
95cf        }
b6b8        Tree_Chain_Division::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);
5d03             printf("%d\n",Tree_Chain_Division::get_sum(1,x));
8e2e         }else{
0f8b             int x,y;
a9b3             scanf("%d%d",&x,&y);
5431             Tree_Chain_Division::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>
421c using namespace std;
5cad typedef long long LL;
40fb const int maxn = 25e4+100;
b1ec const LL INF = 0x3f3f3f3f3f3f3f3fLL;
58a9 int first[maxn],des[maxn*2],nxt[maxn*2],tot;
35b8 int n,m;
667a LL dp[maxn],leng[maxn*2], len[maxn];
e55b int vis[maxn],dep[maxn],fa[maxn];
21fe int sz[maxn],wson[maxn],ttop[maxn],tfa[maxn];int k,h[maxn];
0a19 int stk[maxn],top;int l[maxn],r[maxn],dfs_clock;
a50a inline 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 }
827d void 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 }
4707 void 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 }
620b int 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 }
4ac9 bool cmp(int x,int y){return l[x]<l[y];}
9627 void 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    //
427e    // Created by calabash_boy on 18-6-18.
427e    //
427e
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};
427e
3f9e        const db PI = acosl(-1.0);
427e
2b5b        void ensure_base(type nbase) {
1af7            if (nbase <= base) {
4f2d                return;
95cf            }
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        }
427e
3548        void fft(vector<cp> &a, type n = -1) {
4bae            if (n == -1) {
1528                n = a.size();
95cf            }
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    }
427e
fbc2    vector<cp> fa, fb;
427e
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    }
427e
```

```
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;
ea9c        cp r2(0, -1);
563e        cp r3(ratio, 0);
fb2c        cp r4(0, -ratio);
7e13        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        }
eb13        fft(fa, sz);
e06b        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    }
427e
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;
85f0 int a[maxn],sum[maxn];
6ece int 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(){
9959    scanf("%d%d",&n,&x);
0fe6    cnt[0]=1;
6dbf    for (int i=1;i<=n;i++){
60cb        scanf("%d",a+i);
9a8f        sum[i] =sum[i-1];
5a5e        if(a[i]<x){
f3df            sum[i]++;
95cf        }
6210        cnt[sum[i]]++;
95cf    }
bf61    A.resize(n*2+2);
f81b    B.resize(n*2+2);
    for (int i=0;i<=n;i++){                                          0423
        A[n+i] = cnt[i];                                             6785
        B[n-i] = cnt[i];                                             f450
    }                                                                95cf
    C = fft::multiply(A,B);                                          284a
    C[n*2]-=n+1;                                                     7aa5
    C[n*2]>>=1;                                                      f49a
    for (int i=n*2;i<=n*3;i++){                                      003d
        cout<<C[i]<<"␣";                                             060d
    }                                                                95cf
    return 0;                                                        7021
}                                                                    95cf
```

7.2 FWT

```
//                                                                   427e
// Created by calabash_boy on 18-8-17.                              427e
//                                                                   427e
                                                                     427e
//UOJ 310                                                           427e
#include<bits/stdc++.h>                                              302f
using namespace std;                                                 421c
typedef long long LL;                                                5cad
const int N = 1048576;;                                              a923
const int MOD = 998244353;                                           5bf2
const int INV2 = (MOD+1)>>1;                                         2003
const int INV4 = 1LL*INV2*INV2%MOD;                                  4d4d
int a[N];                                                            ac9d
int n;                                                               5c83
//xor fwt : A[i] = \sigma{-1^([i&j])*a[j]}   [x]:count of 1-bit      427e
void FWT(int *a,int n,int r){                                        3284
    for (int i=1;i<n;i<<=1){                                         65de
        for (int j=0;j<n;j+=(i<<1)){                                 2d6f
            for (int k =0;k<i;k++){                                  3d77
                int x = a[j+k];                                      bf2b
                int y = a[j+k+i];                                    24a0
                if (r){                                              f418
                    a[j+k] = (x+y)%MOD;                              a62b
                    a[j+k+i] = (x-y+MOD)%MOD;                        df0f
                }else{                                               8e2e
                    a[j+k] = 1LL*(x+y)*INV2%MOD;                     a36d
                    a[j+k+i] = 1LL*(x-y+MOD)*INV2%MOD;               5b23
                }                                                    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++){
3c9e         int x;
ea4e         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);
1f14         if (cnt1&1){
243b             a[i] = MOD-a[i];
95cf         }
95cf     }
e16f     FWT(a,N,0);
369d     printf("%d\n",(a[0]+MOD-1)%MOD);
7021     return 0;
95cf }
```

7.3 BerlekampMassey

```
427e //
427e // Created by calabash_boy on 18-8-16.
427e //
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--)
421c using namespace std;
5cad typedef long long LL;
7c77 typedef vector<LL> V;
427e
```

```
b575 const int MOD = 1e9+7;
427e
427e // k 为 m 最高次数 且  a[m] == 1
70d2 namespace 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    }
```

```
    void sample();                                                          bb1a
}                                                                           95cf
void BerlekampMassey::sample(){                                             f425
    V x(6);                                                                 3ddb
    x[0] = 1;                                                               a54e
    x[1] = 2;                                                               989f
    x[2] = 21;                                                              5e15
    x[3] = 212;                                                             5ea7
    x[4] = 2141;                                                            3adf
    x[5] = 21622;                                                           1579
    V a = BerlekampMassey::BM(x);                                           6243
    cout<<"a[n]=";                                                          a849
    for (int i = 0;i<a.size()-2;i++){                                       0126
        cout<<a[i]<<"*a[n-"<<a.size()-1-i<<"]+";                            844c
    }                                                                       95cf
    cout<<a[a.size()-2]<<"*a[n-1]"<<endl;                                   e0ba
}                                                                           95cf
int main(){                                                                 3117
    BerlekampMassey::sample();                                              47ff
    return 0;                                                               7021
}                                                                           95cf
```