

```

1 //START97 TO START67
2 //SMILEY
3 #include<bits/stdc++.h>
4 #include<ext/pb_ds/assoc_container.hpp>
5 #include<ext/pb_ds/tree_policy.hpp>
6
7 using namespace std;
8 using namespace __gnu_pbds;
9
10 template<class T>
11 using oset = tree<T, null_type, less<T>, rb_tree_tag, tree_order_statistics_node_update>;
12 // order_of_key(a) -> gives index of the element(number of elements smaller than a)
13 // find_by_order(a) -> gives the element at index a
14 #define accelerate ios_base::sync_with_stdio(false);cin.tie(NULL);cout.tie(NULL)
15 #define int long long int
16 #define ld long double
17 #define mod1 998244353
18 #define endl "\n"
19 #define ff first
20 #define ss second
21 #define all(x) (x).begin(), (x).end()
22 #define ra(arr,n) vector<int> arr(n);for(int in = 0; in < n; in++) {cin >> arr[in];}
23
24 const int mod = 1e9 + 7;
25 const int inf = 1e18;
26 int MOD(int x) {int a1 = (x % mod); if (a1 < 0) {a1 += mod;} return a1;}
27 int power( int a, int b) {
28     int p = 1; while (b > 0) {if (b & 1)p = (p * a); a = (a * a) ; b >>= 1;}
29     return p;
30 }
31
32 void lessgoo()
33 {
34     int n;
35     cin >> n;
36     string s;
37     cin >> s;
38     string ans = "";
39     ans += s[0];
40     for (int i = 1; i < s.size(); i++) {
41         if (s[i] != ans.back()) {
42             ans += s[i];
43         }
44     }
45     int cnt = 0;
46     if (ans.size() < 3) {
47         cout << 0 << endl;
48         return;
49     }
50     // cout << ans << endl;
51     for (int i = 0; i <= ans.size() - 3; i++) {
52         if (ans[i] == ':' && ans[i + 2] == ':' and ans[i + 1] == ')')cnt++;
53     }
54     cout << cnt << endl;
55 }
56 signed main()
57 {
58     accelerate;
59
60 #ifndef ONLINE_JUDGE
61     freopen("input.txt", "r", stdin);
62     freopen("output.txt", "w", stdout);
63 #endif
64
65
66
67     int test = 1;
68     cin >> test;
69     for (int tcase = 1; tcase <= test; tcase++)

```

```

70     {
71         // cout << "Case #" << tcase << ": ";
72         lessgoo();
73     }
74 }
75 return 0;
76 }
77
78 //NOPALINDROME
79 #include <iostream>
80 #include <string>
81 #include <set>
82 #include <map>
83 #include <stack>
84 #include <queue>
85 #include <vector>
86 #include <utility>
87 #include <iomanip>
88 #include <sstream>
89 #include <bitset>
90 #include <cstdlib>
91 #include <iterator>
92 #include <algorithm>
93 #include <cstdio>
94 #include <cctype>
95 #include <cmath>
96 #include <math.h>
97 #include <ctime>
98 #include <cstring>
99 #include <unordered_set>
100 #include <unordered_map>
101 #include <cassert>
102 #define int long long int
103 #define pb push_back
104 #define mp make_pair
105 #define mod 1000000007
106 #define vl vector<ll>
107 #define all(c) (c).begin(), (c).end()
108 using namespace std;
109
110 const int N=500023;
111 bool vis[N];
112 vector<int> adj[N];
113 long long readInt(long long l, long long r, char endd){
114     long long x=0;
115     int cnt=0;
116     int fi=-1;
117     bool is_neg=false;
118     while(true){
119         char g=getchar();
120         if(g=='-'){
121             assert(fi==-1);
122             is_neg=true;
123             continue;
124         }
125         if('0'<=g && g<='9'){
126             x*=10;
127             x+=g-'0';
128             if(cnt==0){
129                 fi=g-'0';
130             }
131             cnt++;
132             assert(fi!=0 || cnt==1);
133             assert(fi!=0 || is_neg==false);
134
135             assert(!(cnt>19 || (cnt==19 && fi>1)));
136         } else if(g==endd){
137             if(is_neg){
138                 x=-x;

```

```

139     }
140
141     if(!(l <= x && x <= r))
142     {
143         cerr << l << ' ' << r << ' ' << x << '\n';
144         assert(1 == 0);
145     }
146
147     return x;
148 } else {
149     assert(false);
150 }
151 }
152 }
153 string readString(int l,int r,char endd){
154     string ret="";
155     int cnt=0;
156     while(true){
157         char g=getchar();
158         assert(g!=-1);
159         if(g==endd){
160             break;
161         }
162         cnt++;
163         ret+=g;
164     }
165     assert(l<=cnt && cnt<=r);
166     return ret;
167 }
168 long long readIntSp(long long l,long long r){
169     return readInt(l,r,' ');
170 }
171 long long readIntLn(long long l,long long r){
172     return readInt(l,r,'\n');
173 }
174 string readStringLn(int l,int r){
175     return readString(l,r,'\n');
176 }
177 string readStringSp(int l,int r){
178     return readString(l,r,' ');
179 }
180
181 void solve()
182 {
183     int n = readInt(2,1000000000,' ');
184     int k = readInt(1,n-1,'\n');
185     int q = k + 2 + (k-1)/2;
186     int ans = 3*(n/q);
187     if(n%q > k+1){
188         ans += 3;
189     } else if(n%q){
190         ans += 1;
191     }
192     cout << ans;
193 }
194 int32_t main()
195 {
196     #ifndef ONLINE_JUDGE
197     freopen("input.txt", "r", stdin);
198     freopen("output.txt", "w", stdout);
199     #endif
200     ios_base::sync_with_stdio(false);
201     cin.tie(NULL),cout.tie(NULL);
202     int T=readInt(1,5000,'\n');
203     while(T--){
204         solve();
205         cout<<'\n';
206     }
207     assert(getchar()=-1);

```

```

208     cerr << "Time : " << 1000 * ((double)clock()) / (double)CLOCKS_PER_SEC << "ms\n";
209 }
210
211 //MAKESPECIAL
212 #include "bits/stdc++.h"
213 using namespace std;
214
215 #define int long long
216
217 void anubhav(){
218     int n, q;
219     cin >> n >> q;
220
221     int cnt = log2l(n) + 1;
222     int x = (1LL << cnt);
223
224     while(q--){
225         int l, r;
226         cin >> l >> r;
227
228         int rr = r / x;
229         int ll = l / x - (l % x == 0);
230
231         int ans = 2 * (rr - ll);
232         int u = ((l + x - 1) / x) * x;
233         int v = ((r + x - 1) / x) * x;
234
235         if(u - 1 < l || (u) == n) ans--;
236         if(v - 1 <= r && v - 1 >= l && v > r) ans++;
237
238         cout << max(0LL, ans) << "\n";
239     }
240 }
241
242 signed main(){
243     ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0);
244
245     int T = 1;
246     cin >> T;
247     while(T--) anubhav();
248
249     return 0;
250 }
251
252 //TRIPLETMIN
253 #include<bits/stdc++.h>
254 using namespace std;
255
256 #define mod 1000000007
257 typedef set<string> ss;
258 typedef vector<int> vs;
259 typedef map<int, char> msi;
260 typedef pair<int, int> pa;
261 typedef long long int ll;
262
263 ll n, q, i, a[300005], cnt[300005], k;
264 int main()
265 {
266     ios_base::sync_with_stdio(false);
267     cin.tie(0);
268 #ifndef ONLINE_JUDGE
269     freopen("inputf.in", "r", stdin);
270     // freopen("output.txt", "w", stdout);
271 #endif
272
273     int t;
274     cin >> t;
275     while (t--)
276     {

```

```

277     cin >> n >> q;
278     for (i = 0; i < n; i++)
279         cin >> a[i];
280     sort(a, a + n);
281     for (i = 0; i < n; i++)
282     {
283         cnt[i] = (n - i - 1) * (n - i - 2) / 2;
284         if (i > 0)
285             cnt[i] += cnt[i - 1];
286     }
287     while (q--)
288     {
289         cin >> k;
290         cout << a[lower_bound(cnt, cnt + n, k) - cnt] << "\n";
291     }
292 }
293
294 return 0;
295 }
296
297 //SWAPNUM31
298 #include <bits/stdc++.h>
299 #include <ext/pb_ds/assoc_container.hpp>
300 #include <ext/pb_ds/tree_policy.hpp>
301 #include <ext/pb_ds/detail/standard_policies.hpp>
302 using namespace std;
303 using namespace __gnu_pbds;
304 #define ll long long
305 #define int long long
306 #define endl "\n"
307 #define fi first
308 #define se second
309 #define fastio ios_base::sync_with_stdio(false); cin.tie(NULL); cout.tie(NULL);
310 #define fr(a,b,c) for(int a=b; a<c; a++)
311 #define frr(a,b,c) for(int a=b; a>=c; a--)
312 #define pb push_back
313 #define pii pair<int,int>
314 #define R(a) ll a; cin >> a;
315 #define RS(a) string a; cin >> a;
316 typedef tree<long long,null_type,greater_equal<long long>,rb_tree_tag,
tree_order_statistics_node_update> ordered_set;
317 typedef tree<long long,null_type,less<long long>,rb_tree_tag,
tree_order_statistics_node_update> ordered_set1;
318 #define RA(a, n) ll a[(n) + 1] = {}; fr(i, 1, (n)+1) {cin >> a[i];}
319 #define RM(a, n, m) int a[n + 1][m + 1] = {}; fr(i, 1, n+1) {fr(j, 1, m+1) cin >>
a[i][j];}
320 #define reset(X) memset(X, 0, sizeof(X))
321 using vi=vector<int>;
322
323 void __print(long x) {cerr << x;}
324 void __print(long long x) {cerr << x;}
325 void __print(unsigned x) {cerr << x;}
326 void __print(unsigned long x) {cerr << x;}
327 void __print(unsigned long long x) {cerr << x;}
328 void __print(float x) {cerr << x;}
329 void __print(double x) {cerr << x;}
330 void __print(long double x) {cerr << x;}
331 void __print(char x) {cerr << '\\' << x << '\\';}
332 void __print(const char *x) {cerr << '\"' << x << '\"';}
333 void __print(const string &x) {cerr << '\"' << x << '\"';}
334 void __print(bool x) {cerr << (x ? "true" : "false");}
335
336 template<typename T, typename V>
337 void __print(const pair<T, V> &x) {cerr << '{'; __print(x.first); cerr << ','; __print(x
.second); cerr << '}';}
338 template<typename T>
339 void __print(const T &x) {int f = 0; cerr << '{'; for (auto &i: x) cerr << (f++ ? ", " :
""), __print(i); cerr << "}";}
340 void __print() {cerr << "]"<<endl;}

```

```

341 template <typename T, typename... V>
342 void print(T t, V... v) {_print(t); if (sizeof...(v)) cerr << ", "; _print(v...);}
343 #ifndef ONLINE_JUDGE
344 #define deb(x...) cerr << "[" << #x << "]" = ["; _print(x)
345 #else
346 #define deb(x...)
347 #endif
348 #define all(x) (x).begin(), (x).end()
349 const int inf = 1e18;
350 const int mod=998244353;
351 unsigned int power(int x, unsigned int y, int p)
352 {
353     int res = 1;
354     x = x % p;
355     while (y > 0)
356     {
357         if (y & 1)
358             res = (res*x) % p;
359         y = y>>1;
360         x = (x*x) % p;
361     }
362     return res;
363 }
364 int modInverse(int n, int p)
365 {
366     return power(n, p-2, p);
367 }
368
369 void solve () {
370     int n, k;
371     cin >> n >> k;
372     int a[n+1];
373     for( int i = 1; i <= n; i++) {
374         cin >> a[i];
375     }
376     vector<int>v;
377     vector<bool>can_swap(n+1);
378     for(int i = 1; i <= n; i++) {
379         if(i + k <= n) {
380             can_swap[i] = 1;
381             can_swap[i+k] = 1;
382         }
383     }
384
385     for(int i = 1; i <= n; i++) {
386         if(can_swap[i]) {
387             v.push_back(a[i]);
388         }
389     }
390     sort(v.begin(), v.end());
391     int pos = 0;
392     for(int i = 1; i <= n; i++) {
393         if(can_swap[i]) {
394             a[i] = v[pos];
395             pos += 1;
396         }
397     }
398
399     for(int i = 1; i <= n; i++) {
400         cout << a[i] << " ";
401     }
402     cout << endl;
403 }
404
405
406 signed main()
407 {
408     fastio;
409

```

```

410     #ifndef ONLINE_JUDGE
411         if(fopen("input9.txt", "r"))
412         {
413             freopen("input9.txt", "r", stdin);
414             freopen("output9.txt", "w", stdout);
415         }
416     #endif
417     int t=1;
418     cin>>t;
419     //precompute();
420     for(int i = 1; i<=t; i++)
421     {
422         solve();
423     }
424 }
425
426
427 //REMOVESTONES
428 #include<bits/stdc++.h>
429 using namespace std;
430 void solve(){
431     int n, k; cin >> n >> k;
432     int x = k, sqK = sqrt(k);
433     if (k == 1){
434         cout << "Alice" << endl;
435         cout << n << endl;
436         cin >> x;
437         return;
438     }
439     for (int i = 2; i <= sqK; i++) if (k % i == 0){
440         x = i;
441         break;
442     }
443     if (n % x){
444         cout << "Alice" << endl;
445         cout << n % x << endl;
446     }else cout << "Bob" << endl;
447     int ans;
448     while (true){
449         cin >> ans;
450         if (ans == -1) exit(0);
451         if (ans == 0) return;
452         cout << (ans % x ? ans % x : 1) << endl;
453     }
454 }
455 int main(){
456     int test; cin >> test;
457     while (test--) solve();
458 }
459
460 //ZERARR
461 #include<bits/stdc++.h>
462 #define int long long
463 using namespace std;
464
465 void solve()
466 {
467     int n; cin>>n;
468
469     vector<int> a(n);
470     for(int i=0;i<n;i++) cin>>a[i];
471
472     if(n&1)
473     {
474         vector<int> b(n); //b[i] denotes the number of times the move is
475         performed on (i, (i+1)%n)
476         int sum=0;
477         for(int i=0;i<n;i++)
478         {

```

```

478         if(i&1) sum-=a[i];
479         else sum+=a[i];
480     }
481     if(sum<0) cout<<"NO\n";
482     else if(sum%2==1) cout<<"NO\n";
483     else
484     {
485         b[n-1]=sum/2;
486         //b[i]+b[i+1]=a[i+1]
487         for(int i=n-2;i>=0;i--) b[i]=a[i+1]-b[i+1];
488
489         for(int i=0;i<n;i++)
490         {
491             if(b[i]<0)
492             {
493                 cout<<"NO\n";
494                 return;
495             }
496         }
497
498         cout<<"YES\n";
499     }
500 }
501 else
502 {
503     int sumodd=0,sumeven=0;
504     for(int i=0;i<n;i++)
505     {
506         if(i & 1) sumodd+=a[i];
507         else sumeven+=a[i];
508     }
509
510     if(sumodd!=sumeven)
511     {
512         cout<<"NO\n";
513         return;
514     }
515
516     int minn=a[0],maxx=a[0]-a[1];
517     int sum=a[0]-a[1];
518     for(int i=2;i<n;i++)
519     {
520         if(i%2==0)
521         {
522             sum+=a[i];
523             minn=min(minn,sum);
524         }
525         else
526         {
527             sum-=a[i];
528             maxx=max(maxx,sum);
529         }
530     }
531
532     if(minn>=maxx) cout<<"YES\n";
533     else cout<<"NO\n";
534 }
535 }
536
537 signed main()
538 {
539     ios::sync_with_stdio(0);
540     cin.tie(0);
541
542     int t; cin>>t;
543     while(t--)
544     {
545         solve();
546     }

```



```

547 }
548
549 //ABSOLUTEDIFF
550 #include <bits/stdc++.h>
551 using namespace std;
552 vector <pair <int,long long> > g[2010];
553 long long dist[2010];
554 void solve(){
555     int n,m,k;
556     cin>>n>>m>>k;
557     for (int i=0; i<=n; i++) g[i].clear();
558     for (int i=1; i<=m; i++){
559         int pos,val;
560         cin>>pos>>val;
561         g[pos].push_back({0,-val});
562         g[0].push_back({pos,val});
563     }
564     for (int i=1; i<=k; i++){
565         int u,v,d;
566         cin>>u>>v>>d;
567         g[u].push_back({v,d});
568         g[v].push_back({u,d});
569     }
570     for (int i=0; i<=n; i++) dist[i]=0;
571     bool relaxed=0;
572     for (int iters=0; iters<=n; iters++){
573         relaxed=0;
574         for (int i=0; i<=n; i++){
575             for (pair <int,long long> j:g[i]){
576                 if (dist[i]+j.second<dist[j.first]){
577                     dist[j.first]=dist[i]+j.second;
578                     relaxed=1;
579                 }
580             }
581         }
582         if (!relaxed){
583             cout<<"YES\n";
584             return;
585         }
586     }
587     cout<<"NO\n";
588 }
589 int main(){
590     ios_base::sync_with_stdio(0); cin.tie(0);
591     int t; cin>>t;
592     while (t--) solve();
593 }
594
595 //PREFIXES
596 #include <bits/stdc++.h>
597
598 #define el '\n'
599
600 typedef long long ll;
601 typedef long double ld;
602
603 #define Beevo ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0);
604
605 using namespace std;
606
607 const int N = 2e5 + 5, ALPHA = 26, LOG = 20;
608
609 ll oldVal[N];
610 int id, timer, sz[N], trie[N][ALPHA], leaf[N], up[N][LOG], in[N], out[N];
611
612 struct Query {
613     int t, i, k, x;
614     string s;
615 };

```

```

616
617 struct Node {
618     ll sum = 0;
619 };
620
621 struct SegTree {
622     ll lazy[N * 4];
623     Node tree[N * 4];
624     Node neutral = Node();
625
626     Node merge(Node u, Node v) {
627         return {u.sum + v.sum};
628     }
629
630     void propagate(int x, int lX, int rX) {
631         tree[x].sum += lazy[x] * (rX - lX + 1);
632
633         if (lX != rX) {
634             lazy[x * 2] += lazy[x];
635             lazy[x * 2 + 1] += lazy[x];
636         }
637
638         lazy[x] = 0;
639     }
640
641     void update(int x, int lX, int rX, int l, int r, int val) {
642         propagate(x, lX, rX);
643
644         if (lX > r || rX < l)
645             return;
646
647         if (lX >= l && rX <= r) {
648             tree[x].sum += 1LL * (rX - lX + 1) * val;
649
650             if (lX != rX) {
651                 lazy[x * 2] += val;
652                 lazy[x * 2 + 1] += val;
653             }
654
655             return;
656         }
657
658         int m = (lX + rX) >> 1;
659
660         update(x * 2, lX, m, l, r, val);
661         update(x * 2 + 1, m + 1, rX, l, r, val);
662
663         tree[x] = merge(tree[x * 2], tree[x * 2 + 1]);
664     }
665
666     Node query(int x, int lX, int rX, int l, int r) {
667         if (lX > r || rX < l)
668             return neutral;
669
670         propagate(x, lX, rX);
671
672         if (lX >= l && rX <= r)
673             return tree[x];
674
675         int m = (lX + rX) >> 1;
676
677         Node u = query(x * 2, lX, m, l, r);
678         Node v = query(x * 2 + 1, m + 1, rX, l, r);
679
680         return merge(u, v);
681     }
682 } st;
683
684 int insert(int cur, string &s) {

```

```

685     int ch;
686
687     for (auto &i: s) {
688         ch = i - 'a';
689
690         if (!trie[cur][ch])
691             trie[cur][ch] = ++id;
692
693         up[trie[cur][ch]][0] = cur, cur = trie[cur][ch];
694
695         for (int k = 1; k < LOG; k++)
696             up[cur][k] = up[up[cur][k - 1]][k - 1];
697     }
698
699     return cur;
700 }
701
702 int kth(int cur, int k) {
703     for (int i = LOG - 1; i >= 0; i--) {
704         if (k & (1 << i))
705             cur = up[cur][i];
706     }
707
708     return cur;
709 }
710
711 void dfs(int u) {
712     in[u] = timer++;
713
714     for (int i = 0; i < ALPHA; i++) {
715         if (trie[u][i])
716             dfs(trie[u][i]);
717     }
718
719     out[u] = timer - 1;
720 }
721
722 void testCase() {
723     int n;
724     cin >> n;
725
726     string s;
727     for (int i = 0; i < n; i++) {
728         cin >> s;
729
730         sz[i] = s.size();
731         leaf[i] = insert(0, s);
732     }
733
734     int q;
735     cin >> q;
736
737     vector<Query> v;
738     int t, i, k, x, u, cnt = 0;
739     for (int j = 0; j < q; j++) {
740         s.clear();
741
742         cin >> t >> i;
743
744         i--;
745
746         if (t == 1) {
747             cin >> k >> x;
748
749             k--;
750         }
751         else if (t == 2) {
752             cin >> k >> s;
753

```

```

754         k--;
755
756         sz[n + cnt] = k + 1 + s.size();
757         leaf[n + cnt] = insert(kth(leaf[i], sz[i] - k - 1), s);
758
759         cnt++;
760     }
761
762     v.push_back({t, i, k, x, s});
763 }
764
765 dfs(0);
766
767 cnt = 0;
768 for (auto &j: v) {
769     t = j.t, i = j.i, k = j.k, x = j.x, s = j.s;
770
771     if (t == 1) {
772         u = kth(leaf[i], sz[i] - k - 1);
773
774         st.update(1, 0, N - 1, in[u], out[u], x);
775     }
776     else if (t == 2) {
777         oldVal[n + cnt] = st.query(1, 0, N - 1, in[leaf[n + cnt]], in[leaf[n + cnt]]).sum;
778
779         cnt++;
780     }
781     else
782         cout << st.query(1, 0, N - 1, in[leaf[i]], in[leaf[i]]).sum - oldVal[i] <<
783         el;
784 }
785
786 signed main() {
787     Beevo
788
789     int t = 1;
790     // cin >> t;
791
792     while (t--)
793         testCase();
794 }
795
796 //MEX_ARRAY
797 #include <bits/stdc++.h> //Andrei Alexandru a.k.a Sho
798 using ll=long long;
799 using ld=long double;
800 int const INF=1000000005;
801 ll const LINF=1000000000000000005;
802 ll const mod=1e9+7;
803 ld const PI=3.14159265359;
804 ll const NMAX=3e6+5;
805 ld const eps=0.0000001;
806 #pragma GCC optimize("O3")
807 #pragma GCC optimize("Ofast")
808 #define f first
809 #define s second
810 #define pb push_back
811 #define mp make_pair
812 #define endl '\n'
813 #define CODE_START ios_base::sync_with_stdio(false);cin.tie(0);cout.tie(0);
814 using namespace std;
815 ll n,a[200005],cnt[200005],pref[200005];
816 void testcase(){
817     cin>>n;
818     for(ll i=1;i<=n;i++)
819     {
820         cin>>a[i];

```

```

821     cnt[a[i]]++;
822 }
823 pref[0]=cnt[0];
824 for(ll i=1;i<=n;i++)
825 {
826     pref[i]=min(cnt[i],pref[i-1]);
827 }
828 vector<ll>v;
829 ll sum=0;
830 for(ll i=n;i>=0;i--){
831     while(pref[i]-sum>=1){
832         sum++;
833         v.pb(i+1);
834     }
835     cnt[i]-=sum;
836 }
837 for(ll i=1;i<=n;i++)
838 {
839     while(cnt[i]){
840         v.pb(0);
841         cnt[i]--;
842     }
843 }
844 cout<<v.size()<<endl;
845 for(auto it : v){
846     cout<<it<<' ';
847 }
848 cout<<endl;
849 for(ll i=0;i<=n;i++)
850 {
851     cnt[i]=0;
852     pref[i]=0;
853 }
854 }
855 int32_t main(){
856 CODE_START;
857 #ifdef LOCAL
858 freopen("input.in", "r", stdin);
859 #endif
860 ll t=1;
861 cin>>t;
862 while(t--){
863     testcase();
864 }
865 }
866
867 //REMOVEMUL
868
869 #include <iostream>
870 #include <map>
871 #include <vector>
872 using namespace std;
873
874 int main() {
875     int T;
876     cin >> T;
877     while (T--) {
878         int N, M;
879         cin >> N >> M;
880         vector<int> Q(M);
881         long long s = 0;
882         for (int& i : Q)
883             cin >> i, s += i;
884         long long ans = (N * 1LL * (N + 1)) / 2;
885         ans -= s;
886         cout << ans << '\n';
887     }
888     return 0;
889 }

```

```

890
891 //PARTITION
892 #include <map>
893 #include <set>
894 #include <cmath>
895 #include <ctime>
896 #include <queue>
897 #include <stack>
898 #include <cstdio>
899 #include <cstdlib>
900 #include <vector>
901 #include <cstring>
902 #include <algorithm>
903 #include <iostream>
904 using namespace std;
905 typedef double db;
906 typedef long long ll;
907 typedef unsigned long long ull;
908 const int N=1000010;
909 const int LOGN=28;
910 const ll TMD=0;
911 const ll INF=2147483647LL*2147483647LL;
912 int T,n;
913 int a[N];
914 ll dp[N];
915
916 struct Data
917 {
918     ll num,val;
919
920     Data(ll num,ll val):num(num),val(val) {}
921
922     friend bool operator<(Data x,Data y)
923     {
924         if(x.val!=y.val) return x.val>y.val;
925         return x.num<y.num;
926     }
927 };
928
929 struct nod
930 {
931     int l,r;
932     ll mx;
933     nod *lc,*rc;
934 };
935
936 struct Segtree
937 {
938     nod *root;
939
940     Segtree()
941     {
942         build(&root,1,n);
943     }
944
945     void newnod(nod **p,int L,int R)
946     {
947         *p=new nod;
948         (*p)->l=L; (*p)->r=R;
949         (*p)->mx=-INF;
950         (*p)->lc=(*p)->rc=NULL;
951     }
952
953     void build(nod **p,int L,int R)
954     {
955         newnod(p,L,R);
956         if(L==R) return ;
957         int M=(L+R)>>1;
958         build(&(*p)->lc,L,M);

```

```

959         build(&(*p)->rc,M+1,R);
960     }
961
962     void insert(int pos,ll val)
963     {
964         _insert(root,pos,val);
965     }
966
967     void _insert(nod *p,int pos,ll val)
968     {
969         if(p->l==p->r)
970         {
971             p->mx=val;
972             return ;
973         }
974         int M=(p->l+p->r)>>1;
975         if(pos<=M) _insert(p->lc,pos,val);
976         else _insert(p->rc,pos,val);
977         p->mx=max(p->lc->mx,p->rc->mx);
978     }
979
980     ll getmax(int L,int R)
981     {
982         if(L>R) return 0;
983         return _getmax(root,L,R);
984     }
985
986     ll _getmax(nod *p,int L,int R)
987     {
988         if(p->l==L&&p->r==R) return p->mx;
989         int M=(p->l+p->r)>>1;
990         if(R<=M) return _getmax(p->lc,L,R);
991         else if(L>M) return _getmax(p->rc,L,R);
992         else return max(_getmax(p->lc,L,M),_getmax(p->rc,M+1,R));
993     }
994 };
995
996 int main()
997 {
998     scanf("%d",&T);
999     while(T--)
1000     {
1001         scanf("%d",&n);
1002         for(int i=1;i<=n;i++) scanf("%d",&a[i]);
1003         Segtree ST;
1004         set<Data> S;
1005         stack<int> stk;
1006         for(int i=1;i<=n;i++)
1007         {
1008             while((!stk.empty())&&a[stk.top()]<=a[i])
1009             {
1010                 S.erase(Data(stk.top(),dp[stk.top()]));
1011                 stk.pop();
1012             }
1013             if(!stk.empty()) dp[i]=max(S.begin()->val,ST.getmax(stk.top(),i-1)-a[i]+1);
1014             else dp[i]=max((ll)-a[i]+1,ST.getmax(1,i-1)-a[i]+1);
1015             ST.insert(i,dp[i]);
1016             stk.push(i);
1017             S.insert(Data(i,dp[i]));
1018         }
1019         printf("%lld\n",dp[n]);
1020     }
1021
1022     return 0;
1023 }
1024
1025 //ALTTAB
1026 ///(sol_stl.cpp) Expected: AC.
1027 #include <bits/stdc++.h>

```

```

1028 #define maxs 45
1029
1030 using namespace std;
1031 char s[maxs+5];
1032
1033 int main() {
1034     ios::sync_with_stdio(false); cin.tie(0); cout.tie(0);
1035
1036     int n; cin >> n;
1037     stack<string> stk;
1038     string s;
1039
1040     while (n--) {
1041         cin >> s; stk.push(s);
1042
1043     }
1044
1045     set<string> ss;
1046     while (!stk.empty()) {
1047         if (!ss.count(stk.top())) {
1048             cout << stk.top().substr(stk.top().size() - 2, 2);
1049             ss.insert(stk.top());
1050         }
1051
1052         stk.pop();
1053     }
1054
1055     return 0;
1056 }
1057
1058 //GUESS_
1059 #include <map>
1060 #include <set>
1061 #include <cmath>
1062 #include <ctime>
1063 #include <queue>
1064 #include <stack>
1065 #include <cstdio>
1066 #include <cstdlib>
1067 #include <vector>
1068 #include <cstring>
1069 #include <algorithm>
1070 #include <iostream>
1071 using namespace std;
1072 typedef double db;
1073 typedef long long ll;
1074 typedef unsigned long long ull;
1075 const int N=10000010;
1076 const int LOGN=28;
1077 const ll TMD=1000000007;
1078 const ll INF=2147483647;
1079 int T,n;
1080 int pfac[N],f[N],fa[N],fb[N],Sa[N],Sb[N];
1081
1082 void init()
1083 {
1084     for(int i=2;i<N;i++)
1085     {
1086         if(pfac[i]) continue;
1087         for(int j=2;j*i<N;j++) pfac[j*i]=i;
1088     }
1089     for(int i=2;i<N;i++)
1090     {
1091         if(pfac[i]) f[i]=f[i/pfac[i]]+1;
1092         else f[i]=1;
1093     }
1094     for(int i=1;i<N;i++)
1095     {
1096         fa[i]=2*((f[i]+1)/2)+1;

```



```

1097         fb[i]=2*(f[i]/2+1);
1098         Sa[i]=Sa[i-1]+fa[i];
1099         Sb[i]=Sb[i-1]+fb[i];
1100     }
1101 }
1102
1103 ll pw(ll x,ll p)
1104 {
1105     if(!p) return 1;
1106     ll y=pw(x,p>>1);
1107     y=y*y%TMD;
1108     if(p&1) y=y*(x%TMD)%TMD;
1109     return y;
1110 }
1111
1112 ll inv(ll x)
1113 {
1114     return pw(x,TMD-2);
1115 }
1116
1117 int main()
1118 {
1119     init();
1120     scanf("%d",&T);
1121     while(T--)
1122     {
1123         scanf("%d",&n);
1124         int sqn=(int)sqrt(n),cur;
1125         ll P=(-Sa[sqn]-Sb[sqn]+TMD*2)%TMD,Q=TMD-n;
1126         for(int i=1;i<=sqn;i++)
1127         {
1128             P=(P+fa[i]*(n/i))%TMD;
1129             P=(P+fb[i]*(n/i))%TMD;
1130             Q=(Q+n/i)%TMD;
1131         }
1132         cur=sqn+1;
1133         while(cur<=n)
1134         {
1135             int L=cur,R=n+1,M;
1136             while(L+1!=R)
1137             {
1138                 M=(L+R)>>1;
1139                 if(n/M==n/cur) L=M;
1140                 else R=M;
1141             }
1142             P=(P+(Sa[L]-Sa[cur-1])*(n/cur))%TMD;
1143             P=(P+(Sb[L]-Sb[cur-1])*(n/cur))%TMD;
1144             Q=(Q+(L-cur+1)*(n/cur))%TMD;
1145             cur=L+1;
1146         }
1147         Q=Q*2%TMD;
1148         printf("%lld\n",P*inv(Q)%TMD);
1149     }
1150
1151     return 0;
1152 }
1153
1154 //BOX95
1155 #include <bits/stdc++.h>
1156 using namespace std;
1157
1158 #define int long long
1159
1160 int32_t main() {
1161     ios::sync_with_stdio(false);
1162     cin.tie(nullptr);
1163
1164     int t; cin >> t;
1165     while (t--) {

```

```

1166         int n;
1167         cin >> n;
1168
1169         vector<int> a(n);
1170         for (int& p : a) cin >> p;
1171
1172         const auto get_max = [&](int num) -> int {
1173             for (int bit = 60; bit >= 0; bit--) {
1174                 if ((1ll << bit) & num) return bit;
1175             }
1176             assert(false);
1177             return -1;
1178         };
1179
1180         int mx = 0;
1181         for (const int& p : a)
1182             mx = max(mx, get_max(p));
1183
1184         int cnt = 0;
1185         for (const int& p : a)
1186             if ((p & (1ll << mx)))
1187                 ++cnt;
1188
1189         cout << ((cnt + 1) >> 1) << '\n';
1190     }
1191
1192     return 0;
1193 }
1194
1195 //MOONSOON
1196
1197 #include <bits/stdc++.h>
1198 using namespace std;
1199
1200 int32_t main()
1201 {
1202     int t;
1203     cin>>t;
1204     while(t--)
1205     {
1206         int n, m, h;
1207         cin>>n>>m>>h;
1208         vector <int> a(n), b(m);
1209         for(int i=0;i<n;i++)
1210             cin>>a[i];
1211         for(int i=0;i<m;i++)
1212             cin>>b[i];
1213         sort(a.begin(), a.end(), greater <int> ());
1214         sort(b.begin(), b.end(), greater <int> ());
1215         long long sum=0;
1216         for(int i=0;i<min(n, m);i++)
1217             sum+=min(1ll*a[i], 1ll*b[i]*h);
1218         cout<<sum<<"\n";
1219     }
1220 }
1221
1222 //ARRAY_BREAK
1223
1224 //clear adj and visited vector declared globally after each test case
1225 //check for long long overflow
1226 //Mod wale question mein last mein if dalo ie. Ans<0 then ans+=mod;
1227 //Incse of close mle change language to c++17 or c++14
1228 //Check ans for n=1
1229 #pragma GCC target ("avx2")
1230 #pragma GCC optimize ("O3")
1231 #pragma GCC optimize ("unroll-loops")
1232 #include <bits/stdc++.h>
1233 #include <ext/pb_ds/assoc_container.hpp>
1234 #define int long long

```

```

1235 #define IOS std::ios::sync_with_stdio(false);
1236 cin.tie(NULL);cout.tie(NULL);cout.precision(dbl::max_digits10);
1237 #define mod 1000000007ll //99824435311
1238 #define lld long double
1239 #define mii map<int, int>
1240 #define pii pair<int, int>
1241 #define ll long long
1242 #define ff first
1243 #define ss second
1244 #define all(x) (x).begin(), (x).end()
1245 #define rep(i,x,y) for(int i=x; i<y; i++)
1246 #define fill(a,b) memset(a, b, sizeof(a))
1247 #define vi vector<int>
1248 #define setbits(x) __builtin_popcountll(x)
1249 #define print2d(dp,n,m) for(int i=0;i<=n;i++){for(int j=0;j<=m;j++)cout<<dp[i][j]<<"
";cout<<"\n";}
1250 typedef std::numeric_limits< double > dbl;
1251 using namespace __gnu_pbds;
1252 using namespace std;
1253 typedef tree<int, null_type, less<int>, rb_tree_tag, tree_order_statistics_node_update>
indexed_set;
1254 //member functions :
1255 //1. order_of_key(k) : number of elements strictly lesser than k
1256 //2. find_by_order(k) : k-th element in the set
1257 const long long N=200005, INF=2000000000000000000;
1258 const int inf=2e9 + 5;
1259 lld pi=3.1415926535897932;
1260 int lcm(int a, int b)
1261 {
1262     int g=__gcd(a, b);
1263     return a/g*b;
1264 }
1265 int power(int a, int b, int p)
1266 {
1267     if(a==0)
1268         return 0;
1269     int res=1;
1270     a%=p;
1271     while(b>0)
1272     {
1273         if(b&1)
1274             res=(1ll*res*a)%p;
1275         b>>=1;
1276         a=(1ll*a*a)%p;
1277     }
1278     return res;
1279 }
1280 mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());
1281
1282 int getRand(int l, int r)
1283 {
1284     uniform_int_distribution<int> uid(l, r);
1285     return uid(rng);
1286 }
1287 const int MOD=mod;
1288 struct Mint {
1289     int val;
1290
1291     Mint(long long v = 0) {
1292         if (v < 0)
1293             v = v % MOD + MOD;
1294         if (v >= MOD)
1295             v %= MOD;
1296         val = v;
1297     }
1298
1299     static int mod_inv(int a, int m = MOD) {
1300         int g = m, r = a, x = 0, y = 1;

```

```

1301     while (r != 0) {
1302         int q = g / r;
1303         g %= r; swap(g, r);
1304         x -= q * y; swap(x, y);
1305     }
1306     return x < 0 ? x + m : x;
1307 }
1308 explicit operator int() const {
1309     return val;
1310 }
1311 Mint& operator+=(const Mint &other) {
1312     val += other.val;
1313     if (val >= MOD) val -= MOD;
1314     return *this;
1315 }
1316 Mint& operator-=(const Mint &other) {
1317     val -= other.val;
1318     if (val < 0) val += MOD;
1319     return *this;
1320 }
1321 static unsigned fast_mod(uint64_t x, unsigned m = MOD) {
1322     #if !defined(_WIN32) || defined(_WIN64)
1323         return x % m;
1324     #endif
1325     unsigned x_high = x >> 32, x_low = (unsigned) x;
1326     unsigned quot, rem;
1327     asm("divl %4\n"
1328         : "=a" (quot), "=d" (rem)
1329         : "d" (x_high), "a" (x_low), "r" (m));
1330     return rem;
1331 }
1332 Mint& operator*=(const Mint &other) {
1333     val = fast_mod((uint64_t) val * other.val);
1334     return *this;
1335 }
1336 Mint& operator/=(const Mint &other) {
1337     return *this *= other.inv();
1338 }
1339 friend Mint operator+(const Mint &a, const Mint &b) { return Mint(a) += b; }
1340 friend Mint operator-(const Mint &a, const Mint &b) { return Mint(a) -= b; }
1341 friend Mint operator*(const Mint &a, const Mint &b) { return Mint(a) *= b; }
1342 friend Mint operator/(const Mint &a, const Mint &b) { return Mint(a) /= b; }
1343 Mint& operator++() {
1344     val = val == MOD - 1 ? 0 : val + 1;
1345     return *this;
1346 }
1347 Mint& operator--() {
1348     val = val == 0 ? MOD - 1 : val - 1;
1349     return *this;
1350 }
1351 // friend Mint operator<=(const Mint &a, const Mint &b) { return (int)a <= (int)b; }
1352 Mint operator++(int32_t) { Mint before = *this; ++*this; return before; }
1353 Mint operator--(int32_t) { Mint before = *this; --*this; return before; }
1354 Mint operator-() const {
1355     return val == 0 ? 0 : MOD - val;
1356 }
1357 bool operator==(const Mint &other) const { return val == other.val; }
1358 bool operator!=(const Mint &other) const { return val != other.val; }
1359 Mint inv() const {
1360     return mod_inv(val);
1361 }
1362 Mint power(long long p) const {
1363     assert(p >= 0);
1364     Mint a = *this, result = 1;
1365     while (p > 0) {
1366         if (p & 1)
1367             result *= a;
1368         a *= a;
1369     }

```

```

1370         p >>= 1;
1371     }
1372     return result;
1373 }
1374 friend ostream& operator << (ostream &stream, const Mint &m) {
1375     return stream << m.val;
1376 }
1377 friend istream& operator >> (istream &stream, Mint &m) {
1378     return stream>>m.val;
1379 }
1380 };
1381
1382 Mint dp[500][500];
1383 int32_t main()
1384 {
1385     IOS;
1386     int t;
1387     cin>>t;
1388     while(t--)
1389     {
1390         int n, k;
1391         cin>>n>>k;
1392         int a[n];
1393         rep(i,0,n)
1394             cin>>a[i];
1395         string s;
1396         cin>>s;
1397         rep(i,0,n)
1398         {
1399             rep(j,0,n)
1400                 dp[i][j]=0;
1401         }
1402         dp[0][n-1]=1;
1403         Mint inv[n+1];
1404         inv[0]=1;
1405         for(int i=1;i<=n;i++)
1406             inv[i]=(Mint)1/i;
1407         for(int i=1;i<=k;i++)
1408         {
1409             if(s[i-1]=='L')
1410             {
1411                 for(int l=0;l<n;l++)
1412                 {
1413                     Mint temp=dp[l][n-1];
1414                     for(int r=n-1;r>l;r--)
1415                     {
1416                         dp[l][r]-=temp;
1417                         Mint temp2=dp[l][r-1];
1418                         dp[l][r-1]+=((temp*inv[r-l])+dp[l][r]);
1419                         temp=temp2;
1420                     }
1421                 }
1422             }
1423             else
1424             {
1425                 for(int r=0;r<n;r++)
1426                 {
1427                     Mint temp=dp[0][r];
1428                     for(int l=0;l<r;l++)
1429                     {
1430                         dp[l][r]-=temp;
1431                         Mint temp2=dp[l+1][r];
1432                         dp[l+1][r]+=((temp*inv[r-l])+dp[l][r]);
1433                         temp=temp2;
1434                     }
1435                 }
1436             }
1437         }
1438         Mint ans=0;

```

```

1439         for(int i=0;i<n;i++)
1440         {
1441             Mint sum=0;
1442             for(int j=i;j<n;j++)
1443             {
1444                 sum+=a[j];
1445                 ans+=(sum*dp[i][j]);
1446             }
1447         }
1448         cout<<ans<<"\n";
1449     }
1450 }
1451
1452 //CAMOG
1453 #include <bits/stdc++.h>
1454 #include <ext/pb_ds/assoc_container.hpp>
1455 #include <ext/pb_ds/tree_policy.hpp>
1456 using namespace std;
1457 using namespace __gnu_pbds;
1458
1459 #define ll long long int
1460 #define pb push_back
1461 #define all(x) x.begin(),x.end()
1462 #define Max 1000000000000000
1463
1464 template <typename T>
1465 using ordered_set = tree<T, null_type, less<T>, rb_tree_tag,
tree_order_statistics_node_update>;
1466 template <typename T>
1467 using min_heap=priority_queue<T, vector<T>, greater<T>>;
1468
1469 bool chk[500001];
1470 vector<ll> pc[500001];
1471 ll n,dp[500001],mod=1000000007;
1472
1473 ll bigmod(ll v,ll p){
1474     if(!p) return 1;
1475     if(p%2) return (v*bigmod(v,p-1))%mod;
1476     ll temp=bigmod(v,p/2);
1477     return (temp*temp)%mod;
1478 }
1479
1480 int main()
1481 {
1482     //freopen("out/k.in","r",stdin);
1483     //freopen("out/t.out","w",stdout);
1484     scanf("%lli",&n);
1485
1486     for(ll i=1;i<=n;i++) pc[i].pb(1);
1487
1488     for(ll i=2;i<=n;i++){
1489         if(chk[i]) continue;
1490         for(ll j=i;j<=n;j+=i){
1491             chk[j]=1;
1492             ll temp=pc[j].size();
1493             for(ll k=0;k<temp;k++){
1494                 pc[j].pb(-i*pc[j][k]);
1495             }
1496         }
1497     }
1498
1499     for(ll i=1;i<=n;i++){
1500         if(i!=1){
1501             dp[i]=(dp[i]+n)%mod;
1502             dp[i]=(dp[i]*bigmod(n-n/i,mod-2))%mod;
1503         }
1504         for(ll j=2*i;j<=n;j+=i){
1505             ll d=n/i;
1506             ll cp=0;

```

```

1507         for (ll k=0;k<pc[j/i].size();k++) {
1508             cp+=d/pc[j/i][k];
1509         }
1510         //cout<<j<<" "<<n<<" "<<i<<" "<<cp<<endl;
1511         dp[j]=(dp[j]+cp*dp[i])%mod;
1512     }
1513     printf("%lli ",dp[i]);
1514 }
1515 printf("\n");
1516
1517     return 0;
1518 }
1519
1520 //CS2023_404
1521 #include <bits/stdc++.h>
1522 using namespace std;
1523 #define int long long int
1524 #define vi vector<int>
1525 #define rep(i,a,b) for(int i=a;i<b;i++)
1526 #define all(a) a.begin(),a.end()
1527 #define endl "\n"
1528
1529 int mod=1e9+7;
1530 int _pow(int a,int p=mod-2){
1531     if(p<0) return 0;
1532     int res=1;
1533     while(p>0){
1534         if(p&1) res=(res*a)%mod;
1535         p=p>>1; a=(a*a)%mod;
1536     }
1537     return res;
1538 }
1539 void solve(){
1540     int n;
1541     cin>>n;
1542     string str;
1543     cin>>str;
1544     int ls=0,l4=0,l0=0;
1545     int rs=0,r4=0,r0=0;
1546     rep(i,0,n){
1547         rs+=str[i]=='*';
1548         r4+=str[i]=='4';
1549         r0+=str[i]=='0';
1550     }
1551     int ans=0;
1552     rep(i,0,n){
1553         rs-=(str[i]=='*');
1554         r4-=(str[i]=='4');
1555         r0-=(str[i]=='0');
1556
1557         if(str[i]=='0' || str[i]=='*'){
1558             //4 0 4
1559             ans+=(l4*r4)%mod*_pow(2,rs+ls);
1560             ans%=mod;
1561
1562             //4 0 *
1563             ans+=(l4*rs)%mod*_pow(2,rs+ls-1);
1564             ans%=mod;
1565
1566             //* 0 4
1567             ans+=(ls*r4)%mod*_pow(2,rs+ls-1);
1568             ans%=mod;
1569
1570             //* 0 *
1571             ans+=(ls*rs)%mod*_pow(2,rs+ls-2);
1572             ans%=mod;
1573         }
1574         ls+=str[i]=='*';
1575         l4+=str[i]=='4';

```

```

1576         l0+=str[i]=='0';
1577     }
1578     cout<<ans<<endl;
1579 }
1580 int32_t main() {
1581     auto begin = std::chrono::high_resolution_clock::now();
1582     ios_base::sync_with_stdio(false);
1583     cin.tie(0); cout.tie(0);
1584
1585     #ifndef ONLINE_JUDGE
1586     freopen("D:/Desktop/Test_CPP/CS2023_404/CS2023_404_0.in", "r", stdin);
1587     freopen("D:/Desktop/Test_CPP/CS2023_404/CS2023_404_0.out", "w", stdout);
1588     #endif
1589
1590     int t=1;
1591     cin>>t;
1592     while(t--){
1593         solve();
1594
1595         auto end = std::chrono::high_resolution_clock::now();
1596         auto elapsed = std::chrono::duration_cast<std::chrono::nanoseconds>(end - begin);
1597         cerr << "Time measured: " << elapsed.count() * 1e-6 << "ms\n";
1598         return 0;
1599     }
1600
1601     //PLANUM
1602     #include<bits/stdc++.h>
1603     #define int long long
1604     #define mod 998244353
1605     using namespace std;
1606
1607     int binpow(int a,int b){
1608         int res=1;
1609         while(b>0){
1610             if(b&1)res*=a;
1611             a*=a;
1612             res%=mod;
1613             a%=mod;
1614             b>>=1;
1615         }
1616         return res;
1617     }
1618
1619     signed main()
1620     {
1621         ios_base::sync_with_stdio(false);
1622         cin.tie(NULL);
1623         int t;cin>>t;
1624         while(t--){
1625             int n,m;
1626             cin>>n>>m;
1627             if(m>=2*n){
1628                 cout<<-1<<endl;
1629                 continue;;
1630             }
1631             //Formula is  $4 * (2^{m-1} / 3^{n+1}) * ((3/4)^{\max(1, m-n+1)} * (1 - (3/4)^{((m-1)/2 - \max(1, m-n+1) + 1)}))$ 
1632             int ans=4;
1633             ans=(ans*binpow(2,m-1))%mod;
1634             ans=(ans*binpow(binpow(3,n+1),mod-2))%mod;
1635             int minn=max(1ll,m-n);
1636             ans=(ans*binpow(3,minn))%mod;
1637             ans=(ans*binpow(binpow(4,minn),mod-2))%mod;
1638             int len=(m-1)/2-minn+1;
1639             if(len<0){
1640                 cout<<-1<<endl;
1641                 continue;
1642             }
1643             ans=(ans*((1+mod-(binpow(3,len)*binpow(binpow(4,len),mod-2))%mod)%mod))%mod;
1644             if(m<=n)ans+=(binpow(binpow(3,n),mod-2)*binpow(2,m-1))%mod;

```



```

1645         //This is the case when we place nothing on one of the stacks.
1646         cout<<(ans*2)%mod<<endl;
1647     }
1648 }
1649
1650 //VOWMTRX
1651 #include <bits/stdc++.h>
1652 using namespace std;
1653 #define ll long long int
1654 const ll M=1e9+7;
1655
1656 int main() {
1657     ll t; cin>>t;
1658     map<char,ll>vow={{'a',1},{'e',1},{'i',1},{'o',1},{'u',1}};
1659     while(t--){
1660         ll n,k; cin>>n>>k;
1661         string s; cin>>s;
1662         ll ans=1;
1663         ll prev=-1;
1664         ll ct=0;
1665         for (int i = 0; i < n; ++i)
1666         {
1667             if(vow[s[i]]){
1668                 if(ct==0){
1669                     if(prev!=-1){
1670                         ans=(ans*(i-prev))%M;
1671                     }
1672                 }
1673                 ct++;
1674                 if(ct==k){
1675                     ct=0;
1676                     prev=i;
1677                 }
1678             }
1679         }
1680         cout<<ans<<endl;
1681     }
1682     return 0;
1683 }
1684
1685 //CS2023_PON
1686 #include <bits/stdc++.h>
1687 using namespace std;
1688 #define int long long int
1689 #define rep(i,a,b) for(int i=a;i<b;i++)
1690
1691 void solve(){
1692     int n,b;
1693     cin>>n>>b;
1694     vector<int> arr(n);
1695     rep(i,0,n) cin>>arr[i];
1696
1697     int res_and=(1LL<<31)-1;
1698     int cnt=0;
1699     rep(i,0,n){
1700         if((arr[i]&b) == b){
1701             res_and = res_and & arr[i];
1702             cnt++;
1703         }
1704     }
1705
1706     if(res_and == b && cnt>0){
1707         cout<<"YES"<<endl;
1708     }else{
1709         cout<<"NO"<<endl;
1710     }
1711 }
1712
1713 int32_t main() {

```

```

1714     auto begin = std::chrono::high_resolution_clock::now();
1715     ios_base::sync_with_stdio(false);
1716     cin.tie(0); cout.tie(0);
1717
1718     #ifndef ONLINE_JUDGE
1719     freopen("D:/Desktop/Test_CPP/CS2023_PON/CS2023_PON_3.in", "r", stdin);
1720     freopen("D:/Desktop/Test_CPP/CS2023_PON/CS2023_PON_3.out", "w", stdout);
1721     #endif
1722
1723     int t=1;
1724     cin>>t;
1725     while(t--){
1726         solve();
1727     }
1728
1729     auto end = std::chrono::high_resolution_clock::now();
1730     auto elapsed = std::chrono::duration_cast<std::chrono::nanoseconds>(end - begin);
1731     cerr << "Time measured: " << elapsed.count() * 1e-6 << "ms\n";
1732     return 0;
1733 }
1734
1735 //GRDXOR
1736 #include <bits/stdc++.h>
1737 #define int long long
1738 #define ll __int128
1739
1740 #define mod (int)(1e9 + 7)
1741 using namespace std;
1742
1743 int n, m;
1744
1745 int solve(vector<int> &rows, vector<int> &cols, vector<int> total, vector<int> &indi)
1746 {
1747     for (int i = 0; i < 32; i++)
1748     {
1749         total[i] -= rows[i] + cols[i];
1750         total[i] += indi[i];
1751     }
1752     int ans = 0;
1753     for (int i = 0; i < 32; i++)
1754     {
1755         if (indi[i])
1756         {
1757             ans += (1ll << i) * (n * m - n - m + 1 - total[i]);
1758         }
1759         else
1760         {
1761             ans += (1ll << i) * (total[i]);
1762         }
1763     }
1764     return ans;
1765 }
1766
1767 void solve()
1768 {
1769     cin >> n >> m;
1770     vector<vector<int>> v(n, vector<int>(m));
1771     vector<vector<int>> rows(n, vector<int>(32)), cols(m, vector<int>(32));
1772     vector<int> total(32);
1773     for (int i = 0; i < n; i++)
1774     {
1775         for (int j = 0; j < m; j++)
1776         {
1777             cin >> v[i][j];
1778             int it = v[i][j];
1779             int tep = 0;
1780             while (it != 0)
1781             {
1782                 if (it & 1)

```

```

1783         {
1784             rows[i][tep] += 1;
1785             cols[j][tep] += 1;
1786             total[tep] += 1;
1787         }
1788         tep++;
1789         it >>= 1;
1790     }
1791 }
1792
1793 int ans = 0;
1794 vector<int> indi(32);
1795 for (int i = 0; i < n; i++)
1796 {
1797     for (int j = 0; j < m; j++)
1798     {
1799         fill(indi.begin(), indi.end(), 0);
1800         int it = v[i][j], tep = 0;
1801         while (it != 0)
1802         {
1803             if (it & 1)
1804             {
1805                 indi[tep] = 1;
1806             }
1807             else
1808                 indi[tep] = 0;
1809             tep++;
1810             it >>= 1;
1811         }
1812         ans = max(ans, solve(rows[i], cols[j], total, indi));
1813     }
1814 }
1815 cout << ans << '\n';
1816 }
1817 signed main()
1818 {
1819     ios_base::sync_with_stdio(false);
1820     cin.tie(NULL);
1821     int n;
1822     cin >> n;
1823     for (int i = 0; i < n; i++)
1824         solve();
1825 }
1826
1827 //RANKQ
1828 #include <bits/stdc++.h>
1829
1830 #include <ext/pb_ds/assoc_container.hpp>
1831 #include <ext/pb_ds/tree_policy.hpp>
1832 using namespace __gnu_pbds;
1833
1834 #define ordered_set tree<pair<int,int>, null_type,less<pair<int,int>>,
1835 rb_tree_tag,tree_order_statistics_node_update>
1836 using namespace std;
1837
1838 mt19937_64 RNG(chrono::steady_clock::now().time_since_epoch().count());
1839
1840 #define int long long
1841 #define pb push_back
1842 #define rep(i,a,b) for(int i = a; i < b; i++)
1843 #define all(x) x.begin(),x.end()
1844 #define in(a) for(int i = 0; i<a.size(); i++) cin>>a[i];
1845 #define out(a) for(int i = 0; i<a.size(); i++) cout<<a[i]<<" ";
1846 #define vi vector<int>
1847 #define sqrt(x) sqrtl(x)
1848 #define ret(a) cout<<a<<"\n"; return
1849
1850 const int T = 10000;

```

```

1851 const int N = 2e5;
1852 const int Q = 2e5;
1853 const int MAX_A = 1e9;
1854 const int MIN_A = -1e9;
1855
1856 int SUM_N = 0;
1857 int SUM_Q = 0;
1858
1859
1860 bool check(int x, int l, int r, int rank){
1861     int left = max(0LL, l - x);
1862     int right = max(0LL, r - x);
1863
1864     if(left + right + 1 <= rank){
1865         return true;
1866     }
1867
1868     return false;
1869 }
1870
1871 void solve(){
1872     int n, q; cin>>n>>q;
1873
1874     SUM_N += n;
1875     SUM_Q += q;
1876
1877     vi a(n);
1878     in(a);
1879
1880     for(int i = 0; i < n; i++){
1881         assert(MIN_A <= a[i] && a[i] <= MAX_A);
1882     }
1883
1884     vector<vector<pair<int,int>>> m(n);
1885
1886     rep(j,0,q){
1887         int i, x; cin>>i>>x;
1888         assert(1 <= i && i <= n);
1889         assert(1 <= x && x <= n);
1890         m[i-1].push_back({x, j});
1891     }
1892
1893     ordered_set pref, suf;
1894     rep(i,0,n){
1895         suf.insert({a[i], n-i});
1896     }
1897
1898     vector<int> ans(q);
1899
1900     rep(i,0,n){
1901         suf.erase({a[i], n-i});
1902
1903         int l = i - pref.order_of_key({a[i], i});
1904         int r = n - i - 1 - suf.order_of_key({a[i], n-i});
1905
1906         for(auto &x: m[i]){
1907             int rank = x.first;
1908             int query = x.second;
1909
1910             int f = 0, s = n;
1911             while(s - f > 1){
1912                 int mid = (s + f)/2;
1913                 if(check(mid, l, r, rank)){
1914                     s = mid;
1915                 }
1916                 else{
1917                     f = mid;
1918                 }
1919             }

```

```

1920         if(check(f, l, r, rank)) s = f;
1921
1922         ans[query] = s;
1923     }
1924
1925     pref.insert({a[i], i});
1926 }
1927
1928     for(int i = 0; i < q; i++) cout<<ans[i]<<"\n";
1929 }
1930
1931 int32_t main() {
1932     auto begin = std::chrono::high_resolution_clock::now();
1933     ios_base::sync_with_stdio(false); cin.tie(0); cout.tie(0);
1934     int t = 1;
1935     cin>>t;
1936     assert(1 <= t && t <= T);
1937     for(int i = 1; i<=t; i++){
1938         // cout<<"Case #"<<i<<": ";
1939         solve();
1940     }
1941     assert(1 <= SUM_N && SUM_N <= N);
1942     assert(1 <= SUM_Q && SUM_Q <= Q);
1943     auto end = std::chrono::high_resolution_clock::now();
1944     auto elapsed = std::chrono::duration_cast<std::chrono::nanoseconds>(end - begin);
1945     cerr << "Time measured: " << elapsed.count() * 1e-6 << "ms\n";
1946     return 0;
1947 }
1948
1949 //STRSORT
1950 #include <bits/stdc++.h>
1951 using namespace std;
1952
1953 const int TOTAL = 5e5;
1954 const int N = 3e5;
1955 const int TC = 5e4;
1956
1957 int SUM = 0;
1958
1959 vector<vector<int>>> ans;
1960 int res;
1961
1962 void Delete(vector<int> a){
1963     // for(int i = 0; i < a.size(); i++) cout<<a[i]<<" ";
1964     // cout<<"\n";
1965     int n = a.size();
1966
1967     int j = n - 1, k = n;
1968     vector<int> c;
1969     int cur_rem = 0;
1970
1971     for(int i = 0; i < n - 2; i++){
1972         if(a[i] > a[n - 2]){
1973             ans.push_back({2, i + 1 - cur_rem, j - cur_rem, k - cur_rem});
1974             cur_rem++;
1975         }
1976         else c.push_back(a[i]);
1977     }
1978
1979     //now all the elements > an-1 are removed
1980
1981     c.push_back(a[n - 2]);
1982     c.push_back(a[n - 1]);
1983
1984     a = c;
1985     int removed = 0;
1986
1987     n = a.size();
1988

```

```

1989     vector<array<int,2>> stk;
1990
1991     for(int i = 0; i < n - 2; i++){
1992         while(stk.size() > 0 && stk.back()[0] > a[i]){
1993             ans.push_back({2, stk.back()[1], i - removed + 1, n - removed});
1994             stk.pop_back();
1995             removed++;
1996         }
1997         stk.push_back({a[i], i - removed + 1});
1998     }
1999
2000     cout<<"YES\n";
2001     cout<<ans.size()<<" "<<res<<"\n";
2002     for(int i = 0; i < ans.size(); i++){
2003         for(int j = 0; j < ans[i].size(); j++)
2004             cout<<ans[i][j]<<" ";
2005         cout<<"\n";
2006     }
2007 }
2008
2009
2010 void solve(){
2011     ans.clear();
2012     int n; cin>>n;
2013     assert(3 <= n && n <= N);
2014
2015     SUM += n;
2016
2017     vector<int> a(n);
2018     for(int i = 0; i < n; i++) cin>>a[i];
2019
2020     set<int> s;
2021
2022     vector<int> m(n + 1);
2023
2024     for(int i = 0; i < n; i++){
2025         assert(a[i] >= 1 && a[i] <= n);
2026         s.insert(a[i]);
2027         m[a[i]] = i;
2028     }
2029
2030     assert(s.size() == n);
2031
2032     if(a[n - 2] < a[n - 1]){
2033         res = 0;
2034         Delete(a);
2035         return;
2036     }
2037
2038     if(a[n - 2] != n){
2039         res = 1;
2040         ans.push_back({1, m[n] + 1, n});
2041         swap(a[m[n]], a[n - 1]);
2042         Delete(a);
2043         return;
2044     }
2045
2046     if(n == 3){
2047         cout<<"NO\n";
2048         return;
2049     }
2050
2051     if(a[n - 3] == 1){
2052         if(a[n - 1] == 2){
2053
2054             // .... x, n
2055             res = 2;
2056             ans.push_back({1, 1, n - 1});
2057             ans.push_back({1, 1, n});

```

```

2058         swap(a[0], a[n - 2]);
2059         swap(a[0], a[n - 1]);
2060         Delete(a);
2061         return;
2062     }
2063
2064     // ... 1, 2, x (x > 2)
2065     res = 1;
2066     ans.push_back({1, m[2] + 1, n - 1});
2067     swap(a[m[2]], a[n - 2]);
2068     Delete(a);
2069     return;
2070 }
2071
2072 if(a[n - 1] != 1){
2073     // .... 1, x (x > 1)
2074     res = 1;
2075     ans.push_back({1, m[1] + 1, n - 1});
2076     swap(a[m[1]], a[n - 2]);
2077     Delete(a);
2078     return;
2079 }
2080
2081 //... 1, x (x > 1)
2082 res = 2;
2083 ans.push_back({1, 1, n - 1});
2084 ans.push_back({1, 1, n});
2085
2086 swap(a[0], a[n - 2]);
2087 swap(a[0], a[n - 1]);
2088
2089 Delete(a);
2090 }
2091
2092 int main(){
2093     // ios_base::sync_with_stdio(false); cin.tie(0); cout.tie(0);
2094     int t; cin>>t;
2095     assert(1 <= t && t <= TC);
2096
2097     while(t--)
2098         solve();
2099
2100     assert(SUM <= TOTAL);
2101 }
2102
2103 //REMOSET
2104 #include <bits/stdc++.h>
2105 #define endl '\n'
2106 #define filein freopen("input20.in", "r", stdin)
2107 #define fileout freopen("output20.out", "w", stdout)
2108 #define fast ios_base::sync_with_stdio(false); cin.tie(NULL)
2109 using namespace std;
2110 mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());
2111 const int mx=2e5+9;
2112 const int mod=1e9+7;
2113 int main()
2114 {
2115     fast;
2116     int t; cin>>t;
2117     assert(t<=10000);
2118     int total=0;
2119     while(t--)
2120     {
2121         int n; cin>>n;
2122         total+=n;
2123         int odd=0,even=0;
2124         for(int i=0;i<n;i++)
2125         {
2126             int x; cin>>x;

```

```

2127         assert(x>=0 and x<=200000);
2128         if(x%2) odd++;
2129         else even++;
2130     }
2131     long long cnt=1;
2132     for(int i=1;i<=even;i++) cnt=(cnt*2)%mod;
2133     if(odd) cout<<cnt<<endl;
2134     else cout<<cnt-1<<endl;
2135 }
2136 assert(total<=200000);
2137 }
2138
2139 //PALIXOR
2140 #include<bits/stdc++.h>
2141 using namespace std;
2142 vector<int>allPalindrome;
2143 #define fast ios_base::sync_with_stdio(false);cin.tie(NULL)
2144 bool isPalindrome(int num){
2145     int rev=0;
2146     int temp=num;
2147     while (num>0){
2148         rev=rev*10 + (num%10);
2149         num/=10;
2150     }
2151     return (rev==temp);
2152 }
2153 void solve(){
2154     long long int n;
2155     cin>>n;
2156     long long int v[n];
2157     long long int freq[(1<<17)];
2158     memset(freq,0,sizeof(freq));
2159     for (int i=0;i<n;i++){
2160         cin>>v[i];
2161         freq[v[i]]++;
2162     }
2163     long long int ans=0;
2164     for (int i=0;i<allPalindrome.size();i++){
2165         long long int num=allPalindrome[i];
2166         for (int j=0;j<n;j++){
2167             ans+=freq[v[j]^num];
2168         }
2169     }
2170
2171     //Divide ans by 2 as each pair will be calculated twice
2172     ans/=2;
2173     //add all pairs such that A[i]^A[j] = 0 means A[i] and A[j] are same
2174     for (int i=0;i<(1<<17);i++){
2175         ans = ans + (freq[i] * (freq[i]+1))/2;
2176     }
2177     cout<<ans<<endl;
2178 }
2179
2180 void generatePalindromes(){
2181     for (int i=1;i<=(1<<17);i++){
2182         if (isPalindrome(i)){
2183             allPalindrome.push_back(i);
2184         }
2185     }
2186 }
2187
2188 int main(){
2189     int t;
2190     fast;
2191     cin>>t;
2192     generatePalindromes();
2193     while (t--){
2194         solve();
2195     }

```



```

2196     }
2197 }
2198
2199 //K_POWGAME
2200             //Enjoying CP as always!
2201             // NO FAREWELL: KEEP GOOGLE COMPETITIONS ALIVE
2202 #include <bits/stdc++.h>
2203 using namespace std;
2204 #define int long long
2205
2206 signed main() {
2207     //freopen("small_tests_input.txt", "r", stdin);
2208     //freopen("small_tests_output.txt", "w", stdout);
2209     ios_base::sync_with_stdio(false);
2210     cin.tie(NULL);
2211     int t;
2212     cin>>t;
2213     while(t--) {
2214         int n,k;
2215         cin>>n>>k;
2216         int z = n%(k+1);
2217         if((z&1LL) || z==k) {
2218             cout<<"Shivansh\n";
2219         } else {
2220             cout<<"Tina\n";
2221         }
2222     }
2223     return 0;
2224 }
2225
2226 //MAIL_DELIVER
2227 #include <bits/stdc++.h> //Andrei Alexandru a.k.a Sho
2228 using ll=long long;
2229 using ld=long double;
2230 int const INF=1000000005;
2231 ll const LINF=1000000000000000005;
2232 ll const mod=1000000007;
2233 ld const PI=3.14159265359;
2234 ll const MAX_N=3e5+5;
2235 ld const EPS=0.00000001;
2236 #pragma GCC optimize("O3")
2237 #pragma GCC optimize("Ofast")
2238 #define f first
2239 #define s second
2240 #define pb push_back
2241 #define mp make_pair
2242 #define endl '\n'
2243 #define sz(a) (int)a.size()
2244 #define CODE_START ios_base::sync_with_stdio(false);cin.tie(0);cout.tie(0);
2245 using namespace std;
2246 ll t,n,m,k,x[20005],d[20005],viz[20005];
2247 vector<ll>g[10005];
2248 void testcase(){
2249     cin>>n>>m>>k;
2250     for(ll i=1;i<=n;i++)
2251     {
2252         viz[i]=0;
2253     }
2254     for(ll i=1;i<=k;i++)
2255     {
2256         cin>>x[i];
2257     }
2258     priority_queue<pair<ll,ll>,vector<pair<ll,ll>>,greater<pair<ll,ll>>>q;
2259     for(ll i=1;i<=k;i++)
2260     {
2261         cin>>d[i];
2262         viz[x[i]]=min(-d[i],viz[x[i]]);
2263         q.push(mp(-d[i],x[i]));
2264     }

```

```

2265     for(ll i=1;i<=m;i++)
2266     {
2267         ll x,y;
2268         cin>>x>>y;
2269         g[x].pb(y);
2270         g[y].pb(x);
2271     }
2272     while(!q.empty()){
2273         ll x,y;
2274         x=q.top().s;
2275         y=q.top().f;
2276         q.pop();
2277         if(viz[x]!=y){
2278             continue;
2279         }
2280         for(auto it : g[x]){
2281             if(viz[x]+1<viz[it]){
2282                 viz[it]=viz[x]+1;
2283                 q.push(mp(viz[it],it));
2284             }
2285         }
2286     }
2287     int ans=1;
2288     for(ll i=1;i<=n;i++)
2289     {
2290         if(viz[i]==0){
2291             ans=0;
2292         }
2293     }
2294     if(ans){
2295         cout<<"YES"<<endl;
2296     }else cout<<"NO"<<endl;
2297     for(ll i=1;i<=n;i++)
2298     {
2299         viz[i]=0;
2300         g[i].clear();
2301     }
2302 }
2303 int32_t main(){
2304     CODE_START;
2305     cin>>t;
2306     while(t--){
2307         testcase();
2308     }
2309 }
2310
2311 //LASTRBS
2312 // author : divyesh_11
2313 #include<bits/stdc++.h>
2314 using namespace std;
2315
2316 #define ll long long
2317
2318 vector<vector<int>> dp;
2319 vector<ll> pre;
2320
2321 int helper(ll i, ll val, ll n, ll size, vector<ll> &a)
2322 {
2323     if (i >= n)
2324     {
2325         return val == 0;
2326     }
2327     if (dp[i][val] != -1)
2328     {
2329         return dp[i][val];
2330     }
2331     int ans1 = 0;
2332     int ans2 = 0;
2333     if (val >= a[i])

```

```

2334     {
2335         ans1 = helper(i + 1, val - a[i], n, size, a);
2336     }
2337     ll can = (size - pre[i] - val) / 2;
2338     if (can >= a[i])
2339     {
2340         ans2 = helper(i + 1, val + a[i], n, size, a);
2341     }
2342     return dp[i][val] = (ans1 || ans2);
2343 }
2344
2345 void solve()
2346 {
2347     ll n;
2348     cin >> n;
2349
2350     string s;
2351     cin >> s;
2352
2353     if (n & 1)
2354     {
2355         cout << "NO" << endl;
2356         return;
2357     }
2358
2359     vector<ll> a;
2360     ll count = 1;
2361     char c = s[0];
2362     for (int i = 1; i < n; i++)
2363     {
2364         if (s[i] == c)
2365         {
2366             count++;
2367         }
2368         else
2369         {
2370             a.push_back(count);
2371             count = 1;
2372             c = s[i];
2373         }
2374     }
2375     a.push_back(count);
2376     ll size = a.size();
2377     pre.clear();
2378     pre.resize(size + 1, 0);
2379     dp.clear();
2380     dp.resize(size + 1, vector<int>(n + 1, -1));
2381     pre[1] = a[0];
2382     for (int i = 2; i <= size; i++)
2383     {
2384         pre[i] = pre[i - 1] + a[i - 1];
2385     }
2386     if (helper(0, 0, size, n, a))
2387     {
2388         cout << "yes" << endl;
2389         ll curr = 0;
2390         ll i = 0;
2391         string ans;
2392         while (i < size - 1)
2393         {
2394             if (dp[i + 1][curr + a[i]] == 1)
2395             {
2396                 ll count = a[i];
2397                 while (count--)
2398                 {
2399                     ans.push_back('(');
2400                 }
2401                 curr += a[i];
2402             }

```

```

2403         else
2404         {
2405             ll count = a[i];
2406             while (count--)
2407             {
2408                 ans.push_back(' ');
2409             }
2410             curr -= a[i];
2411         }
2412         i++;
2413     }
2414     while (a[i]--)
2415     {
2416         ans.push_back(' ');
2417     }
2418     cout << ans << endl;
2419 }
2420 else
2421 {
2422     cout << "no" << endl;
2423 }
2424 }
2425
2426 int main()
2427 {
2428     ios_base :: sync_with_stdio(false);
2429     cin.tie(NULL);
2430     cout.tie(NULL);
2431
2432     int t = 1;
2433     cin >> t;
2434
2435     for (int i = 1; i <= t; i++)
2436     {
2437         solve();
2438     }
2439
2440     return 0;
2441 }
2442
2443 //SORTP9
2444 // #pragma GCC optimize("O3,unroll-loops")
2445 // #pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
2446 #include "bits/stdc++.h"
2447 using namespace std;
2448 using ll = long long int;
2449 mt19937_64 rng(chrono::high_resolution_clock::now().time_since_epoch().count());
2450
2451 int main()
2452 {
2453     ios::sync_with_stdio(false); cin.tie(0);
2454
2455     const int bits = 20;
2456     const int full = (1 << bits) - 1;
2457     int n; cin >> n;
2458     vector<int> a(n);
2459     for (int &x : a) cin >> x;
2460
2461     vector<int> indices(n), start(1 << bits, -1);
2462     for (int i = 0; i < n; ++i) indices[i] = i;
2463     sort(begin(indices), end(indices), [&] (int i, int j) {return a[i] < a[j];});
2464     for (int i = 0; i < n; ++i) {
2465         if (i > 0 and a[indices[i]] == a[indices[i-1]]) continue;
2466         start[a[indices[i]]] = i;
2467     }
2468
2469     basic_string<int> val, comp, z, cont;
2470     int Time, ncomps;
2471     auto dfs = [&] (const auto &self, int u, auto &f) -> int {

```

```

2472         int low = val[u] = ++Time, x; z.push_back(u);
2473
2474     if (u <= full) {
2475         for (int bit = 0; bit < bits; ++bit) if (u & (1 << bit)) {
2476             int e = u ^ (1 << bit);
2477             if (comp[e] < 0) low = min(low, val[e] ?: self(self,e,f));
2478         }
2479         if (start[u] != -1) {
2480             int v = start[u], value = a[indices[v]];
2481             while (v < n) {
2482                 int e = indices[v];
2483                 if (a[e] != value) break;
2484                 e += 1 << bits;
2485                 if (comp[e] < 0) low = min(low, val[e] ?: self(self,e,f));
2486                 ++v;
2487             }
2488         }
2489     }
2490     else {
2491         int e = a[u - full - 1] ^ full;
2492         if (comp[e] < 0) low = min(low, val[e] ?: self(self,e,f));
2493     }
2494
2495     if (low == val[u]) {
2496         do {
2497             x = z.back(); z.pop_back();
2498             comp[x] = ncomps;
2499             cont.push_back(x);
2500         } while (x != u);
2501         f(cont); cont.clear();
2502         ncomps++;
2503     }
2504     return val[u] = low;
2505 };
2506 auto SCC = [&] (auto f) {
2507     int N = n + (1 << bits);
2508     val.assign(N, 0); comp.assign(N, -1);
2509     Time = ncomps = 0;
2510     for (int i = 0; i < N; ++i) if (comp[i] < 0) dfs(dfs, i, f);
2511 };
2512
2513 vector<int> ans(n);
2514 SCC([&] (const auto &v) {
2515     vector<int> indices, values;
2516     for (int x : v) if (x > full) {
2517         indices.push_back(x-1 - full);
2518         values.push_back(a[x-1-full]);
2519     }
2520     if (indices.empty()) return;
2521
2522     sort(begin(indices), end(indices));
2523     sort(begin(values), end(values));
2524     int sz = indices.size();
2525     for (int i = 0; i < sz; ++i) ans[indices[i]] = values[i];
2526 });
2527 for (int x : ans) cout << x << ' ';
2528 cout << '\n';
2529 }
2530
2531 //FLRCNT
2532
2533 #include <bits/stdc++.h>
2534
2535 #define el '\n'
2536
2537 typedef long long ll;
2538 typedef long double ld;
2539
2540 #define Beevo ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0);

```

```

2541
2542 using namespace std;
2543
2544 const int N = 1e6 + 5, M = 1e9 + 7;
2545
2546 int fact[N], inv[N];
2547
2548 int add(int a, int b) {
2549     b = (b + M) % M;
2550
2551     return (a + b) % M;
2552 }
2553
2554 int mul(int a, int b) {
2555     return 1LL * a * b % M;
2556 }
2557
2558 int modPow(int b, int p) {
2559     if (p == 0)
2560         return 1;
2561
2562     int x = modPow(b, p / 2);
2563
2564     return p % 2 == 0 ? mul(x, x) : mul(b, mul(x, x));
2565 }
2566
2567 int modInvFer(int n) {
2568     return modPow(n, M - 2);
2569 }
2570
2571 void pre() {
2572     fact[0] = 1;
2573
2574     for (int i = 1; i < N; i++)
2575         fact[i] = mul(fact[i - 1], i);
2576
2577     inv[N - 1] = modInvFer(fact[N - 1]);
2578
2579     for (int i = N - 2; i >= 0; i--)
2580         inv[i] = mul(inv[i + 1], i + 1);
2581 }
2582
2583 int ncr(int n, int r) {
2584     return mul(fact[n], mul(inv[r], inv[n - r]));
2585 }
2586
2587 int hockeyStick(int n, int r) {
2588     return ncr(n + 1, r);
2589 }
2590
2591 int sumDiagonal(int n1, int r1, int n2, int r2) {
2592     return add(hockeyStick(n2, r2), -hockeyStick(n1 - 1, r1 - 1));
2593 }
2594
2595 void testCase() {
2596     int n;
2597     cin >> n;
2598
2599     int sum, x1, y1, x2, y2;
2600     for (int i = 1; i <= n; i++) {
2601         sum = 0;
2602
2603         for (int j = 0; j < n; j += i) {
2604             x1 = j, x2 = min(j + i - 1, n - 1);
2605             y1 = j - j / i, y2 = y1 + x2 - x1;
2606
2607             sum = add(sum, sumDiagonal(x1, y1, x2, y2));
2608         }
2609     }

```

```

2610         cout << sum << ' ';
2611     }
2612
2613     cout << el;
2614 }
2615
2616 signed main() {
2617     Beevo
2618
2619     pre();
2620
2621     int t = 1;
2622     cin >> t;
2623
2624     while (t--)
2625         testCase();
2626 }
2627
2628 //MLTDVD
2629 #include <bits/stdc++.h>
2630
2631 #define el '\n'
2632
2633 typedef long long ll;
2634 typedef long double ld;
2635
2636 #define Beevo ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0);
2637
2638 using namespace std;
2639
2640 const int N = 1e7 + 5, M = 1e9 + 7;
2641
2642 int spf[N], maxFreq[N];
2643
2644 int mul(int a, int b) {
2645     return 1LL * a * b % M;
2646 }
2647
2648 int modPow(int b, int p) {
2649     if (!p)
2650         return 1;
2651
2652     int x = modPow(b, p / 2);
2653
2654     return p % 2 == 0 ? mul(x, x) : mul(b, mul(x, x));
2655 }
2656
2657 int modInvFer(int n) {
2658     return modPow(n, M - 2);
2659 }
2660
2661 void sieve() {
2662     for (int i = 1; i < N; i++)
2663         spf[i] = i;
2664
2665     for (int i = 2; i * i < N; i++) {
2666         if (spf[i] == i) {
2667             for (int j = i * 2; j < N; j += i)
2668                 spf[j] = min(spf[j], i);
2669         }
2670     }
2671 }
2672
2673 void factorize(int n) {
2674     int pf, freq;
2675     while (n > 1) {
2676         pf = spf[n], freq = 0;
2677
2678         while (n % pf == 0)

```

```

2679         freq++, n /= pf;
2680
2681         maxFreq[pf] = max(maxFreq[pf], freq);
2682     }
2683 }
2684
2685 void testCase() {
2686     sieve();
2687
2688     int n;
2689     cin >> n;
2690
2691     int g = 0;
2692     for (int i = 0, a; i < n; i++)
2693         cin >> a, g = __gcd(g, a), factorize(a);
2694
2695     int l = 1;
2696     for (int i = 1; i < N; i++)
2697         l = mul(l, modPow(i, maxFreq[i]));
2698
2699     int q;
2700     cin >> q;
2701
2702     int k, p, d = mul(l, modInvFer(g));
2703     while (q--) {
2704         cin >> k;
2705
2706         p = modPow(d, k);
2707
2708         cout << mul(g, p) << ' ' << mul(l, p) << el;
2709     }
2710 }
2711
2712 signed main() {
2713     Beevo
2714
2715     int t = 1;
2716     // cin >> t;
2717
2718     while (t--)
2719         testCase();
2720 }
2721
2722 //PECULIARFUNK
2723 #include<bits/stdc++.h>
2724
2725 // #include <ext/pb_ds/assoc_container.hpp>
2726 // #include <ext/pb_ds/tree_policy.hpp>
2727
2728 using namespace std;
2729 // using namespace __gnu_pbds;
2730
2731 #define int long long
2732 #define all(x) x.begin(), x.end()
2733
2734 // typedef tree<int, null_type, less<int>, rb_tree_tag,
2735 // tree_order_statistics_node_update> oset;
2736 typedef unsigned long long ull;
2737 typedef long double lld;
2738
2739 const int N=1e7+2, M=1e5+10, mod=1e9+7, inf = 4e18, moda = 998244353;
2740 const lld pi = 3.141592653589793;
2741
2742 void solve(){
2743     int n, k;
2744     cin>>n>>k;
2745
2746     vector<int> a(n);
2747     for(int i=0; i<n; i++) cin>>a[i];

```



```

2747     sort(all(a));
2748
2749     if(n == 1){
2750         cout<<0<<'\n';
2751         return;
2752     }
2753
2754     if(n<4){
2755         if(a[0] == a[n-1]){
2756             cout<<k<<'\n';
2757             return;
2758         }
2759         cout<<0<<'\n';
2760         return;
2761     }
2762
2763     int min_count = ceil((lld)(n-4)/3)+2;
2764     int curr_count = unique(all(a))-a.begin();
2765     cout<<k*max(0ll, min_count-curr_count)<<'\n';
2766 }
2767
2768 signed main(){
2769
2770     ios::sync_with_stdio(false);
2771     cin.tie(0);  cout.tie(0);
2772
2773     cout<<setprecision(15)<<fixed;
2774
2775     // freopen("zinput.in","r",stdin);
2776     // freopen("zoutput.out","w",stdout);
2777
2778     int tt;
2779     cin>>tt;
2780
2781     // for(int i=1; i<=tt; i++){
2782     //     cout<<"Case #"<i<<": ";
2783     //     solve();
2784     // }
2785
2786     while(tt--){
2787         solve();
2788     }
2789
2790     // solve();
2791 }
2792
2793 //BEAUTIFULARR
2794 #include<bits/stdc++.h>
2795 #include <ext/pb_ds/assoc_container.hpp>
2796 #include <ext/pb_ds/tree_policy.hpp>
2797 using namespace std;
2798 using namespace __gnu_pbds;
2799
2800 #pragma GCC target ("avx2")
2801 #pragma GCC optimization ("O3")
2802 #pragma GCC optimization ("unroll-loops")
2803 #pragma GCC target("popcnt")
2804
2805 template <typename T>
2806 using ordered_set = tree<T, null_type, less<T>, rb_tree_tag,
2807 tree_order_statistics_node_update>;
2808
2809 template <typename T>
2810 using ordered_multiset = tree<T, null_type, less_equal<T>, rb_tree_tag,
2811 tree_order_statistics_node_update>;
2812 using namespace std;
2813
2814 const double pi = acos(-1);

```

```

2814 // DEBUG FUNCTIONS
2815 #ifndef ONLINE_JUDGE
2816
2817 template<typename T>
2818 void __p(T a) {
2819     cout<<a;
2820 }
2821 template<typename T, typename F>
2822 void __p(pair<T, F> a) {
2823     cout<<"{";
2824     __p(a.first);
2825     cout<<", ";
2826     __p(a.second);
2827     cout<<"}";
2828 }
2829 template<typename T>
2830 void __p(std::vector<T> a) {
2831     cout<<"{";
2832     for(auto it=a.begin(); it<a.end(); it++)
2833         __p(*it), cout<<", }" [it+1==a.end()];
2834 }
2835 template<typename T>
2836 void __p(std::set<T> a) {
2837     cout<<"{";
2838     for(auto it=a.begin(); it!=a.end(); ){
2839         __p(*it);
2840         cout<<", }" [++it==a.end()];
2841     }
2842 }
2843
2844 template<typename T>
2845 void __p(std::multiset<T> a) {
2846     cout<<"{";
2847     for(auto it=a.begin(); it!=a.end(); ){
2848         __p(*it);
2849         cout<<", }" [++it==a.end()];
2850     }
2851 }
2852 template<typename T>
2853 void __p(ordered_set<T> a) {
2854     cout<<"{";
2855     for(auto it=a.begin(); it!=a.end(); ){
2856         __p(*it);
2857         cout<<", }" [++it==a.end()];
2858     }
2859 }
2860
2861 template<typename T>
2862 void __p(ordered_multiset<T> a) {
2863     cout<<"{";
2864     for(auto it=a.begin(); it!=a.end(); ){
2865         __p(*it);
2866         cout<<", }" [++it==a.end()];
2867     }
2868 }
2869 template<typename T, typename F>
2870 void __p(std::map<T,F> a) {
2871     cout<<"{\n";
2872     for(auto it=a.begin(); it!=a.end(); ++it)
2873     {
2874         __p(it->first);
2875         cout << ": ";
2876         __p(it->second);
2877         cout<<"\n";
2878     }
2879     cout << "}\n";
2880 }
2881
2882 template<typename T, typename ...Arg>

```

```

2883 void __p(T a1, Arg ...a) {
2884     __p(a1);
2885     __p(a...);
2886 }
2887 template<typename Arg1>
2888 void __f(const char *name, Arg1 &&arg1) {
2889     cout<<name<<" : ";
2890     __p(arg1);
2891     cout<<endl;
2892 }
2893 template<typename Arg1, typename ... Args>
2894 void __f(const char *names, Arg1 &&arg1, Args &&... args) {
2895     int bracket=0,i=0;
2896     for(;; i++)
2897         if(names[i]==' '&&bracket==0)
2898             break;
2899         else if(names[i]=='(')
2900             bracket++;
2901         else if(names[i]==')')
2902             bracket--;
2903     const char *comma=names+i;
2904     cout.write(names,comma-names)<<" : ";
2905     __p(arg1);
2906     cout<<" | ";
2907     __f(comma+1,args...);
2908 }
2909 #define trace(...) cout<<"Line:"<<__LINE__<<" ", __f(#__VA_ARGS__, __VA_ARGS__)
2910 #else
2911 #define trace(...)
2912 #define error(...)
2913 #endif
2914
2915 // DEBUG FUNCTIONS END
2916
2917 # define FOR(i, a, n) for(int i = a; i<n;i++)
2918 # define FORd(i, a, n) for(int i = a; i >= n; i--)
2919 #define ll long long
2920 ll mod = 1000000007;
2921 # define endl "\n"
2922 # define int ll
2923 # define printArr(arr, n) FOR(abcd,0, n){cout<<arr[abcd]<<" ";}cout<<endl;
2924 #define f first
2925 #define se second
2926 #define pb push_back
2927 #define pob pop_back
2928 #define sz(x) (int)x.size()
2929 #define all(x) x.begin(), x.end()
2930 typedef vector<long long> vi;
2931 typedef pair<long long, long long> pii;
2932 typedef vector<pair<long long, long long>> vpi;
2933 typedef vector<vector<int>> vvi;
2934 int gcdExtended(int a, int b, int* x, int* y)
2935 {
2936     // Base Case
2937     if (a == 0)
2938     {
2939         *x = 0, *y = 1;
2940         return b;
2941     }
2942
2943     int x1, y1; // To store results of recursive call
2944     int gcd = gcdExtended(b % a, a, &x1, &y1);
2945
2946     // Update x and y using results of recursive
2947     // call
2948     *x = y1 - (b / a) * x1;
2949     *y = x1;
2950
2951     return gcd;

```

```

2952 }
2953
2954
2955 // Function to find modulo inverse of a
2956 ll modInverse(ll a, ll m)
2957 {
2958     int x, y;
2959     int g = gcdExtended(a, m, &x, &y);
2960     if (g != 1)
2961         return 0;
2962     else
2963     {
2964         // m is added to handle negative x
2965         ll res = (x % m + m) % m;
2966         return res;
2967     }
2968 }
2969
2970 ll nCr(int n, int r){
2971     if(r>n){
2972         return 0;
2973     }
2974     if(r>n-r){
2975         r = n-r;
2976     }
2977     ll ans = 1;
2978     for(int i = 1; i<=r ; i++){
2979         ans *= (n-i+1);
2980         ans%= mod;
2981         ans *= modInverse(i, mod);
2982         ans %= mod;
2983     }
2984
2985     return ans;
2986 }
2987
2988
2989 ll binpow(ll a, ll b) {
2990     if (b == 0)
2991         return 1;
2992     long long res = binpow(a, b / 2);
2993     if (b % 2)
2994         return (res * res)%mod * a % mod;
2995     else
2996         return (res * res) %mod;
2997 }
2998
2999 // const int Max = 2e5 +1;
3000 // ll fact[Max];
3001 // ll inv_fact[Max];
3002
3003 // void preSolveFact(ll n){
3004 //     ll ans = 1;
3005 //     fact[0] = 1;
3006 //     for(int i = 1; i<=n; i++){
3007 //         ans *=i;
3008 //         ans %= mod;
3009 //         fact[i] = ans;
3010 //     }
3011 //     inv_fact[n] = binpow(fact[n], mod-2);
3012
3013 //     for(int i = n-1; i>=0; i--){
3014 //         inv_fact[i] = inv_fact[i+1] * (i+1) %mod;
3015 //     }
3016 // }
3017 // ll nCr_pre(ll n, ll r){
3018 //     if(n>=r && n>=0 && r>=0)
3019 //         return fact[n] * inv_fact[r] %mod * inv_fact[n-r]%mod;
3020 //     else return 0;

```

```

3021 // }
3022
3023 signed main(){
3024     #ifdef LOCALFLAG
3025         freopen("input.in", "r", stdin);
3026         freopen("output.in", "w", stdout);
3027     #endif
3028     ios_base::sync_with_stdio(false);
3029     cin.tie(NULL);
3030     int t = 1;
3031     cin>>t;
3032     while(t--){
3033         int n, k;
3034         cin>>n>>k;
3035         int sum = 0;
3036         vi arr(n);
3037         FOR(i, 0, n){
3038             cin>>arr[i];
3039         }
3040         sort(all(arr));
3041         int l = arr[0], r = arr[0] + k;
3042         int ans = l;
3043         while(l <= r){
3044             int mid = (l + r)/2;
3045             int ops = 0;
3046             for(int i = 0; i < n; i++){
3047                 ops += max((ll)0, mid - arr[i]);
3048             }
3049             if(ops <= k) {
3050                 l = mid + 1;
3051                 ans = mid;
3052             }
3053             else r = mid - 1;
3054         }
3055         FOR(i, 0, n){
3056             if(arr[i] < ans) {
3057                 k -= ans - arr[i];
3058                 arr[i] = ans;
3059             }
3060         }
3061         FOR(i, 0, k){
3062             arr[i]++;
3063         }
3064         FOR(i, 0, n){
3065             arr[i] %= mod;
3066             sum += arr[i];
3067         }
3068         sum %= mod;
3069         ans = 0;
3070         FOR(i, 0, n){
3071             ans += (sum - arr[i])*arr[i] % mod;
3072             ans = (ans + mod) % mod;
3073         }
3074         ans %= mod;
3075         cout<<ans * modInverse(2, mod) % mod<<endl;
3076     }
3077 }
3078
3079 //DIVIDE_GROUP
3080 #include "bits/stdc++.h"
3081 using namespace std;
3082
3083 // #include <ext/pb_ds/assoc_container.hpp>
3084 // using namespace __gnu_pbds;
3085 // template<class T> using oset = tree<T,null_type,less_equal// for indexed_multiset */
3086 // <T> ,rb_tree_tag,tree_order_statistics_node_update> ; // order_of_key(k) -> # of
3087 // elem strictly < k .
3088 //
3089 // *(s.find_by_order(k)) ->

```

```

element at index K .
3089 #define int long long
3090 using ll= long long;
3091 #define ld long double
3092 #define endl '\n'
3093 #define dbg(x) cout<<#x<<" is -> "<<x<<endl
3094 #define speed_ ios_base::sync_with_stdio(false),cin.tie(0), cout.tie(0)
3095 #define pb push_back
3096 #define po pop_back
3097 #define mp make_pair
3098 #define sab(x) x.begin(),x.end()
3099 #define rsab(x) x.rbegin(),x.rend()
3100 #define ff first
3101 #define ss second
3102 #define sz(x) (int)x.size()
3103 #define sp(x) fixed<<setprecision(x)
3104 #define uni(edge) edge.erase(unique(edge.begin(),edge.end()),edge.end());
3105 #define to_up(x) transform(sab(x),x.begin(),::toupper)
3106 #define to_low(x) transform(x.begin(),x.end(),x.begin(),::tolower)
3107
3108 const int M = 1000000007;
3109 const int MM = 998244353;
3110 const ld Pi= acos(-1);
3111 const int N=1e6+5;
3112 const int inf=1e18;
3113
3114 void simp(){
3115     // dp?, graph?, bs on answer?, compress/sort queries/array?, stupid observation?
3116
3117
3118     int m;
3119     cin>>m;
3120     int n;
3121     cin>>n;
3122     vector<int>a(m);
3123     int sum=0;
3124     for(int i=0;i<m;i++){
3125         cin>>a[i];
3126         sum+=a[i];
3127     }
3128     if(n>m){
3129         cout<<0<<endl;
3130         return ;
3131     }
3132     int l=1;
3133     int r=(sum+n-1)/n;
3134     r+=2;
3135     while(l+1<r){
3136
3137         int mid=(l+r)/2;
3138         int req=mid*n;
3139         for(int i=0;i<m;i++){
3140             req-=min(mid,a[i]);
3141             if(req<=0)break;
3142         }
3143         if(req<=0){
3144             l=mid;
3145         }
3146         else r=mid;
3147     }
3148     cout<<l<<"\n";
3149 }
3150
3151
3152
3153
3154
3155
3156

```

```

3157
3158 signed main(){
3159
3160     speed_;
3161
3162     // freopen("input06.txt", "r", stdin);
3163     // freopen("ouput06.txt", "w", stdout);
3164
3165     int t;
3166     t=1;
3167     cin>>t;
3168
3169
3170     int curr=1;
3171     while(t--){
3172
3173         // cout<<"Case #"<<curr++<<": ";
3174         simp();
3175
3176     }
3177     return 0;
3178 }
3179
3180 //CHEFPOLYGONS
3181 #pragma GCC optimize("Ofast")
3182 #pragma GCC optimize("O3")
3183
3184 #include <bits/stdc++.h>
3185 using namespace std;
3186
3187 #define el '\n'
3188 #define F first
3189 #define S second
3190
3191 typedef long long ll;
3192 typedef long double ld;
3193
3194 bool multipleTestCases = 0, sublime = 1;
3195 const int N = 15000000 + 5;
3196
3197 bool isP[N], isPrime[N];
3198 vector<int> primes;
3199 ll n, mnX = 1e15, mnY = 1e15, mxX, mxY, ans, lowerLimit, upperLimit;
3200 vector<pair<ll, ll>> v;
3201
3202 void sieve() {
3203     isPrime[0] = isPrime[1] = 1;
3204     for (int i = 2; i < N; i++){
3205         isP[i] = isPrime[i] = 1;;
3206     }
3207
3208     for (int i = 2; i * i < N; i++) {
3209         if (isP[i]) {
3210             for (int j = i * i; j < N; j += i) {
3211                 isP[j] = 0;
3212             }
3213         }
3214     }
3215
3216     for (int i = 2; i < N; i++) {
3217         if (isP[i]) {
3218             primes.push_back(i);
3219         }
3220     }
3221 }
3222
3223 void processPrimes() {
3224     lowerLimit = mnX + mnY, upperLimit = mxX + mxY;
3225

```

```

3226     if (upperLimit >= N) {
3227         for (auto &p : primes) {
3228             ll md = lowerLimit % p;
3229             ll start = (md ? lowerLimit + (p - md) : lowerLimit);
3230
3231             if (start == p) {
3232                 start += p;
3233             }
3234
3235             for (ll i = start; i <= upperLimit; i += p) {
3236                 isPrime[i - lowerLimit] = 0;
3237             }
3238         }
3239     }
3240     else {
3241         for (int i = lowerLimit; i <= upperLimit; i++) {
3242             isPrime[i - lowerLimit] = isP[i];
3243         }
3244     }
3245 }
3246
3247 void doWork() {
3248     sieve();
3249
3250     cin >> n;
3251
3252     v.resize(n);
3253
3254     for (auto &i : v) {
3255         cin >> i.F;
3256     }
3257
3258     for (auto &i : v) {
3259         cin >> i.S;
3260
3261         mnX = min(mnX, i.F);
3262         mxX = max(mxX, i.F);
3263         mnY = min(mnY, i.S);
3264         mxY = max(mxY, i.S);
3265     }
3266
3267     processPrimes();
3268
3269     for (int i = 0; i < n; i++) {
3270         pair<ll, ll> cur = v[i], nxt = v[(i + 1) % n];
3271
3272         ll dx = nxt.F - cur.F, dy = nxt.S - cur.S, g = __gcd(abs(dx), abs(dy));
3273         ll xUnit = dx / g, yUnit = dy / g;
3274         ll x = cur.F, y = cur.S;
3275
3276         while (x != nxt.F or y != nxt.S) {
3277             ans += isPrime[x + y - lowerLimit];
3278
3279             x += xUnit;
3280             y += yUnit;
3281         }
3282     }
3283
3284     cout << ans << el;
3285 }
3286
3287 signed main() {
3288     #ifdef ONLINE_JUDGE
3289         ios_base::sync_with_stdio(0), cin.tie(0), cout.tie(0);
3290     #else
3291         if (sublime) {
3292             freopen("input.txt", "r", stdin);
3293             freopen("output.txt", "w", stdout);
3294         }

```



```

3295 #endif
3296     int tests = 1;
3297     if (multipleTestCases) {
3298         cin >> tests;
3299     }
3300     for (int tc = 1; tc <= tests; tc++) {
3301         doWork();
3302     }
3303 }
3304
3305 //CHEFGRAPH
3306 #include <bits/stdc++.h>
3307 #include <ext/pb_ds/assoc_container.hpp>
3308 #include <ext/pb_ds/tree_policy.hpp>
3309 using namespace std;
3310 using namespace __gnu_pbds;
3311 #define int long long int
3312 #define ordered_set tree<int, nuint_type, less<int>,
3313 rb_tree_tag, tree_order_statistics_node_update>
3314 mt19937 rng(std::chrono::duration_cast<std::chrono::nanoseconds>(chrono::
3315 high_resolution_clock::now().time_since_epoch()).count());
3316 #define mp make_pair
3317 #define pb push_back
3318 #define F first
3319 #define S second
3320 const int N=100005;
3321 #define M 1000000007
3322 #define BINF 1e16
3323 #define init(arr, val) memset(arr, val, sizeof(arr))
3324 #define MAXN 10000005
3325 #define deb(xx) cout << #xx << " " << xx << "\n";
3326 const int LG = 22;
3327
3328 bool isPath(vector<int> arr, int n, int u, int v) {
3329     for(auto i : arr) {
3330         if(i < 1 or i > n) return false;
3331         if(i == u or i == v) return false;
3332     }
3333     return true;
3334 }
3335
3336 void solve() {
3337     int n, q;
3338     cin >> n >> q;
3339
3340     if(n <= 7) {
3341         vector<vector<vector<int>>> path(n + 1, vector<vector<int>>>(n + 1));
3342         vector<vector<int>> V(n + 1, vector<int> (n + 1, 0));
3343         for(int i = 1; i <= n; i++) {
3344             for(int j = 1; j <= n; j++) {
3345                 if(i == j) continue;
3346                 vector<int> a;
3347                 int c = 0;
3348                 for(int mask = 1; mask < (1LL << n); mask++) {
3349                     vector<int> arr;
3350                     arr.push_back(i);
3351                     for(int l = 0; l < n; l++) {
3352                         if((mask & (1LL << l)) > 0) {
3353                             if((l + 1) == i or (l + 1) == j) continue;
3354                             arr.push_back(l + 1);
3355                         }
3356                     }
3357                     arr.push_back(j);
3358                     int len = 0;
3359                     for(auto l : arr) {
3360                         len = len ^ l;
3361                     }

```

```

3362         if(len > c) {
3363             c = len;
3364             a = arr;
3365             V[i][j] = len;
3366         } else if (len == c) {
3367             if(arr.size() < a.size()) {
3368                 c = len;
3369                 a = arr;
3370                 V[i][j] = len;
3371             }
3372         }
3373     }
3374     path[i][j] = a;
3375 }
3376 }
3377
3378 while(q-->0) {
3379     int u, v;
3380     cin >> u >> v;
3381     vector<int> arr = path[u][v];
3382
3383     cout << V[u][v] << " " << arr.size() << endl;
3384     for(auto i : arr) {
3385         cout << i << " ";
3386     }
3387     cout << endl;
3388 }
3389
3390 return 0;
3391 }
3392
3393 int x = log2(n);
3394 int len = (1LL << (x + 1)) - 1;
3395
3396 while(q-->0) {
3397     int u, v;
3398     cin >> u >> v;
3399
3400     int u1 = u, v1 = v;
3401     if(u > v) {
3402         swap(u, v);
3403     }
3404
3405     if(n == ((1LL << x) + 1) and u == (1LL << x) and v == ((1LL << x) + 1)) {
3406         cout << (1LL << x) - 1 << " " << 3 << endl;
3407         cout << u1 << " " << ((1LL << x) - 2) << " " << v1 << endl;
3408         continue;
3409     }
3410
3411     int c = u ^ v;
3412     if(c == len) {
3413         cout << (1LL << (x + 1)) - 1 << " " << 2 << endl;
3414         cout << u1 << " " << v1 << endl;
3415         continue;
3416     }
3417
3418     c = len ^ u ^ v;
3419     if(c <= n) {
3420         if(c == u or c == v) {
3421
3422             if(isPath({1, 1 ^ u}, n, u, v)) {
3423                 cout << (1LL << (x + 1)) - 1 << " " << 4 << endl;
3424                 cout << u1 << " " << 1 << " " << (1 ^ u) << " " << v1 << endl;
3425             } else {
3426                 cout << (1LL << (x + 1)) - 1 << " " << 4 << endl;
3427                 cout << u1 << " " << 2 << " " << (2 ^ u) << " " << v1 << endl;
3428             }
3429
3430         } else {

```

```

3431         cout << (1LL << (x + 1)) - 1 << ' ' << 3 << endl;
3432         cout << u1 << ' ' << c << ' ' << v1 << endl;
3433     }
3434
3435     } else {
3436
3437         int r = n - (1LL << x);
3438         if(r >= 2) {
3439
3440             for(int i = 0; i <= 2; i++) {
3441                 if(isPath({(1LL << x) + i, ((1LL << x) + i) ^ c}, n, u, v)) {
3442                     cout << (1LL << (x + 1)) - 1 << ' ' << 4 << endl;
3443                     cout << u1 << ' ' << (1LL << x) + i << ' ' << (((1LL << x) + i)
3444                         ^ c) << ' ' << v1 << endl;
3445                     break ;
3446                 }
3447             }
3448
3449             } else if (r == 1) {
3450
3451                 int f = 0;
3452                 for(int i = 0; i <= 1; i++) {
3453                     if(isPath({(1LL << x) + i, ((1LL << x) + i) ^ c}, n, u, v)) {
3454                         f = 1;
3455                         cout << (1LL << (x + 1)) - 1 << ' ' << 4 << endl;
3456                         cout << u1 << ' ' << (1LL << x) + i << ' ' << (((1LL << x) + i)
3457                             ^ c) << ' ' << v1 << endl;
3458                         break ;
3459                     }
3460                 }
3461
3462                 if(f == 1) {
3463                     continue ;
3464                 }
3465
3466                 cout << (1LL << (x + 1)) - 1 << ' ' << 5 << endl;
3467                 cout << u1 << ' ' << (1LL << x) + 1 << ' ' << 2 << ' ' << (1LL << x) - 3
3468                     << ' ' << v1 << endl;
3469
3470             } else {
3471
3472                 int f = 0;
3473
3474                 if(isPath({1LL << x, (1LL << x) ^ c}, n, u, v)) {
3475                     f = 1;
3476                     cout << (1LL << (x + 1)) - 1 << ' ' << 4 << endl;
3477                     cout << u1 << ' ' << (1LL << x) << ' ' << ((1LL << x) ^ c) << ' ' <<
3478                         v1 << endl;
3479                 }
3480
3481                 if(f == 1) {
3482                     continue;
3483                 }
3484
3485                 if(isPath({1LL << x, 1, 1 ^ u}, n, u, v)) {
3486                     cout << (1LL << (x + 1)) - 1 << ' ' << 5 << endl;
3487                     cout << u1 << ' ' << (1LL << x) << ' ' << 1 << ' ' << (1 ^ u) << ' '
3488                         << v1 << endl;
3489                 } else {
3490                     cout << (1LL << (x + 1)) - 1 << ' ' << 5 << endl;
3491                     cout << u1 << ' ' << (1LL << x) << ' ' << 2 << ' ' << (2 ^ u) << ' '
3492                         << v1 << endl;
3493                 }
3494             }
3495         }
3496     }
3497 }

```

```

3494     }
3495
3496 }
3497
3498
3499 #undef int
3500 int main() {
3501     #define int long long int
3502     ios_base::sync_with_stdio(false);
3503     cin.tie(0);
3504     cout.tie(0);
3505     #ifndef ONLINE_JUDGE
3506         freopen("input.txt", "r", stdin);
3507         freopen("optput.txt", "w", stdout);
3508     #endif
3509
3510     // int T;
3511     // cin >> T;
3512
3513     // for(int tc = 1; tc <= T; tc++){
3514     //     cout << "Case #" << tc << ": ";
3515     //     solve();
3516     // }
3517
3518     return 0;
3519
3520 }
3521
3522 //XOR_EQ_SUM
3523 #include <bits/stdc++.h>
3524 using namespace std;
3525
3526 int main() {
3527     ios::sync_with_stdio(false);
3528     cin.tie(nullptr);
3529     int T;
3530     cin >> T;
3531     assert(1 <= T && T <= 1e5);
3532     while (T--) {
3533         long long l, r;
3534         cin >> l >> r;
3535         assert(0 <= l && l <= r && r <= 1e18);
3536         long long ans = 0;
3537         for (int b = 0; b < 60; b++) {
3538             long long msb = (1LL << b);
3539             if (l <= msb && msb <= r) {
3540                 ans += msb - l;
3541             }
3542         }
3543         if (l == 0) {
3544             ans++;
3545         }
3546         cout << ans << '\n';
3547     }
3548     return 0;
3549 }
3550
3551 //MAXSUMPERM
3552 #include <bits/stdc++.h>
3553 using namespace std;
3554
3555 const int M = 1e9 + 7;
3556
3557 int solve() {
3558     int n, q;
3559     cin >> n >> q;
3560     vector<int> a(n + 1);
3561
3562     for (int i = 1; i <= n; i++) {

```

```

3563         cin >> a[i];
3564     }
3565     sort(a.begin() + 1, a.end());
3566     vector<int> w(n + 2);
3567
3568     while (q--) {
3569         int l, r;
3570         cin >> l >> r;
3571         w[l]++;
3572         w[r + 1]--;
3573     }
3574
3575     for (int i = 1; i <= n; i++) w[i] += w[i - 1];
3576     vector<int> ord(n + 1, 0);
3577     iota(ord.begin(), ord.end(), 0);
3578     sort(ord.begin() + 1, ord.end(), [&](int i, int j) {
3579         return w[i] < w[j];
3580     });
3581     long long ans = 0;
3582     auto b = a;
3583
3584     for (int i = 1; i <= n; i++) {
3585         b[ord[i]] = a[i];
3586         ans += 1LL * w[ord[i]] * a[i];
3587     }
3588     cout << ans << "\n";
3589     for (int i = 1; i <= n; i++) {
3590         cout << b[i] << " ";
3591     }
3592     cout << "\n";
3593     return 0;
3594 }
3595
3596 int main() {
3597     ios_base::sync_with_stdio(0);
3598     cin.tie(0);
3599     cout.tie(0);
3600
3601     int t = 1;
3602     cin >> t;
3603     while (t--) solve();
3604     return 0;
3605 }
3606
3607 //MORE_INV
3608
3609 //clear adj and visited vector declared globally after each test case
3610 //check for long long overflow
3611 //Mod wale question mein last mein if dalo ie. Ans<0 then ans+=mod;
3612 //Inc case of close mle change language to c++17 or c++14
3613 //Check ans for n=1
3614 #pragma GCC target ("avx2")
3615 #pragma GCC optimize ("O3")
3616 #pragma GCC optimize ("unroll-loops")
3617 #include <bits/stdc++.h>
3618 #include <ext/pb_ds/assoc_container.hpp>
3619 #define int long long
3620 #define IOS std::ios::sync_with_stdio(false);
3621 cin.tie(NULL); cout.tie(NULL); cout.precision(dbl::max_digits10);
3622 #define pb push_back
3623 #define mod 1000000007ll //998244353ll
3624 #define lld long double
3625 #define mii map<int, int>
3626 #define pii pair<int, int>
3627 #define ll long long
3628 #define ff first
3629 #define ss second
3630 #define all(x) (x).begin(), (x).end()
3631 #define rep(i,x,y) for(int i=x; i<y; i++)

```

```

3631 #define fill(a,b) memset(a, b, sizeof(a))
3632 #define vi vector<int>
3633 #define setbits(x) __builtin_popcountll(x)
3634 #define print2d(dp,n,m) for(int i=0;i<=n;i++){for(int j=0;j<=m;j++)cout<<dp[i][j]<<"
";cout<<"\n";}
3635 typedef std::numeric_limits< double > dbl;
3636 using namespace __gnu_pbds;
3637 using namespace std;
3638 typedef tree<int, null_type, less_equal<int>, rb_tree_tag,
tree_order_statistics_node_update> indexed_set;
3639 //member functions :
3640 //1. order_of_key(k) : number of elements strictly lesser than k
3641 //2. find_by_order(k) : k-th element in the set
3642 const long long N=600015, INF=2000000000000000000;
3643 const int inf=2e9 + 5;
3644 lld pi=3.1415926535897932;
3645 int lcm(int a, int b)
3646 {
3647     int g=__gcd(a, b);
3648     return a/g*b;
3649 }
3650 int power(int a, int b, int p)
3651 {
3652     if(a==0)
3653         return 0;
3654     int res=1;
3655     a%=p;
3656     while(b>0)
3657     {
3658         if(b&1)
3659             res=(1ll*res*a)%p;
3660         b>>=1;
3661         a=(1ll*a*a)%p;
3662     }
3663     return res;
3664 }
3665 mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());
3666
3667 int getRand(int l, int r)
3668 {
3669     uniform_int_distribution<int> uid(l, r);
3670     return uid(rng);
3671 }
3672
3673 int st[4*N],lazy[4*N],ar[N];
3674
3675 void propagate(int node, int l, int r)
3676 {
3677     if(l!=r)
3678     {
3679         lazy[node*2]+=lazy[node];
3680         lazy[node*2+1]+=lazy[node];
3681     }
3682     st[node]+=lazy[node];
3683     lazy[node]=0;
3684 }
3685 void build(int node, int l, int r)
3686 {
3687     if(l==r)
3688     {
3689         st[node]=0;
3690         lazy[node]=0;
3691         return;
3692     }
3693     int mid=(l+r)/2;
3694     build(node*2, l, mid);
3695     build(node*2+1, mid+1, r);
3696     st[node]=max(st[node*2], st[node*2+1]);
3697     lazy[node]=0;

```

```

3698     return;
3699 }
3700 void update(int node, int l, int r, int x, int y, int val)
3701 {
3702     if(lazy[node]!=0)
3703         propogate(node, l, r);
3704     if(y<x||x>r||y<l)
3705         return;
3706     if(l>=x&&r<=y)
3707     {
3708         st[node]+=val;
3709         if(l!=r)
3710         {
3711             lazy[node*2]+=val;
3712             lazy[node*2+1]+=val;
3713         }
3714         return;
3715     }
3716     int mid=(l+r)/2;
3717     update(node*2, l, mid, x, y, val);
3718     update(node*2+1, mid+1, r, x, y, val);
3719     st[node]=max(st[node*2], st[node*2+1]);
3720     return;
3721 }
3722 int query(int node, int l, int r, int x, int y)
3723 {
3724     if(lazy[node]!=0)
3725         propogate(node, l, r);
3726     if(y<x||y<l||x>r)
3727         return -INF;
3728     if(l>=x&&r<=y)
3729         return st[node];
3730     int mid=(l+r)/2;
3731     return max(query(node*2, l, mid, x, y), query(node*2+1, mid+1, r, x, y));
3732 }
3733
3734 int count_inv(vector <int> a)
3735 {
3736     int n=a.size(), ans=0;
3737     indexed_set s;
3738     for(auto num:a)
3739     {
3740         ans+=s.order_of_key(-num);
3741         s.insert(-num);
3742     }
3743     return ans;
3744 }
3745 void compress(vector <int> &a)
3746 {
3747     set <int> s;
3748     for(auto num:a)
3749         s.insert(num);
3750     mii mp;
3751     int z=2;
3752     for(auto it:s)
3753         mp[it]=z++;
3754     for(auto &num:a)
3755         num=mp[num];
3756 }
3757 int32_t main()
3758 {
3759     IOS;
3760     int t;
3761     cin>>t;
3762     while(t--)
3763     {
3764         int n;
3765         cin>>n;
3766         vector <int> a(n);

```

```

3767         rep(i,0,n)
3768         cin>>a[i];
3769         compress(a);
3770         for(auto &num:a)
3771             num*=2;
3772         int n2=(2*n)+5;
3773         int base=count_inv(a);
3774         build(1, 1, n2);
3775         for(auto num:a)
3776             update(1, 1, n2, num+1, n2, 1);
3777         for(auto num:a)
3778         {
3779             update(1, 1, n2, num+1, n2, -1);
3780             cout<<base+query(1, 1, n2, 1, n2)-query(1, 1, n2, num, num)<<" ";
3781             update(1, 1, n2, 1, num-1, 1);
3782         }
3783         cout<<"\n";
3784     }
3785 }
3786
3787 //CHEAPOFF - ITERATIVE
3788 #include<bits/stdc++.h>
3789 using namespace std;
3790 using ll=long long;
3791
3792 #ifdef ANI
3793 #include "D:/DUSTBIN/local_inc.h"
3794 #else
3795 #define dbg(...) 0
3796 #endif
3797
3798 class Testcase{
3799 public:
3800     ll N,M,K,ans;
3801     vector<vector<ll>> e;
3802     vector<ll> a;
3803 };
3804
3805 ll solution(Testcase T) {
3806     vector<ll> a=T.a;
3807     vector<vector<ll>> edges=T.e;
3808     ll k=T.K;
3809     ll n=a.size();
3810     vector<vector<ll>> e(n);
3811     for(auto x:edges) {
3812         e[x[0]-1].push_back(x[1]-1);
3813         e[x[1]-1].push_back(x[0]-1);
3814     }
3815     vector<ll> init(n); iota(init.begin(),init.end(),0);
3816     vector<vector<ll>> current={init};
3817
3818     function<ll(ll,ll,ll,ll,vector<ll>&,vector<ll>&> bfs=[&](ll root,ll bit,ll is,vector<
ll>&vis,vector<ll>&have)->ll{
3819         if(vis[root] or ((a[root]>>bit)&1)!=is) return 0;
3820         queue<ll> q; q.push(root);
3821         ll res=0; vis[root]=1;
3822         while(!q.empty()) {
3823             ll cur=q.front(); q.pop(); res++; vis[cur]=1;
3824             for(ll node:e[cur]) {
3825                 if(!vis[node] && have[node] && ((a[node]>>bit)&1)==is) {
3826                     q.push(node); vis[node]=1;
3827                 }
3828             }
3829         }
3830         return res;
3831     };
3832
3833     ll ans=(1ll<<32)-1,mx=ans;
3834     for(ll bit=31;bit>=0;bit--) {

```



```

3835         vector<vector<ll>> next;
3836         vector<ll> have(n,0),vis(n,0);
3837
3838         for(vector<ll>&v: current) {
3839             array<vector<ll>,2> c;
3840             for(ll i:v) {
3841                 have[i]=1;
3842                 c[(a[i]>>bit)&1].push_back(i);
3843             }
3844             array<ll,2> sz({0,0});
3845             for(ll z=0;z<2;z++) {
3846                 for(ll i:v) {
3847                     sz[z]=max(sz[z],bfs(i,bit,z,vis,have));
3848                 }
3849             }
3850             for(ll i:v) {
3851                 have[i]=0;
3852                 vis[i]=0;
3853             }
3854             if(sz[0]<k && sz[1]<k) {
3855                 continue;
3856             }
3857             for(ll z=0;z<2;z++) {
3858                 if(sz[z]>=k) {
3859                     ans&=(mx-(1ll<<bit));
3860                     next.push_back(c[z]);
3861                 }
3862             }
3863         }
3864         if(!next.empty())
3865             current=next;
3866     }
3867     return ans==((1ll<<32)-1)?-1:ans;
3868 }
3869
3870 int main() {
3871     int t=1;
3872     cin>>t;
3873     while(t--) {
3874         Testcase T;
3875         cin>>T.N>>T.M>>T.K;
3876         T.a=vector<ll>(T.N);
3877         for(ll i=0;i<T.N;i++)
3878             cin>>T.a[i];
3879         vector<vector<ll>> e;
3880         for(ll i=0;i<T.M;i++) {
3881             ll u,v; cin>>u>>v;
3882             e.push_back({u,v});
3883         }
3884         T.e=e;
3885         cout<<solution(T)<<"\n";
3886     }
3887 }
3888
3889 //CHEAPOFF-RECURSIVE
3890 // #pragma GCC optimize("O3,unroll-loops")
3891 // #pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
3892 #include "bits/stdc++.h"
3893 using namespace std;
3894 using ll = long long int;
3895 mt19937_64 rng(chrono::high_resolution_clock::now().time_since_epoch().count());
3896
3897 int main()
3898 {
3899     ios::sync_with_stdio(false); cin.tie(0);
3900
3901     int t; cin >> t;
3902     while (t--) {
3903         int n, m, k; cin >> n >> m >> k;

```

```

3904     vector<int> a(n);
3905     for (int &x : a) cin >> x;
3906
3907     vector adj(n, vector<int>());
3908     for (int i = 0; i < m; ++i) {
3909         int u, v; cin >> u >> v;
3910         adj[u - 1].push_back(v - 1);
3911         adj[v - 1].push_back(u - 1);
3912     }
3913
3914     vector<int> mark(n);
3915     int ops = 0;
3916     auto bfs = [&] (int src) {
3917         queue<int> q;
3918         q.push(src);
3919         mark[src] = 2;
3920         int sz = 0;
3921         while (!q.empty()) {
3922             int u = q.front(); q.pop();
3923             ++sz;
3924             ops += adj[u].size();
3925             for (int v : adj[u]) {
3926                 if (mark[v] == 1) {
3927                     mark[v] = 2;
3928                     q.push(v);
3929                 }
3930             }
3931         }
3932         return sz;
3933     };
3934
3935     auto solve = [&] (const auto &self, const auto &vals) -> int {
3936         // For this set of values, min cost of having a component of size >= k
3937         // INT_MAX if impossible
3938         if (vals.size() < k) return INT_MAX;
3939
3940         int mxsz = 0;
3941         for (auto &[x, id] : vals) mark[id] = 1;
3942         for (auto [x, id] : vals) {
3943             if (mark[id] == 2) continue;
3944             mxsz = max(mxsz, bfs(id));
3945         }
3946         for (auto &[x, id] : vals) mark[id] = 0;
3947
3948         if (mxsz < k) return INT_MAX;
3949         if (vals[0][0] == vals.back()[0]) return 0;
3950
3951         int rem = vals[0][0] ^ vals.back()[0];
3952         int highbit = 31 - __builtin_clz(rem);
3953         vector<array<int, 2>> left, right;
3954         for (auto &x : vals) {
3955             if (x[0] & (1 << highbit)) {
3956                 right.push_back(x);
3957                 right.back()[0] ^= 1 << highbit;
3958             }
3959             else left.push_back(x);
3960         }
3961         int ret = min(self(self, left), self(self, right));
3962         if (ret < INT_MAX) return ret;
3963         for (auto &x : right) left.push_back(x);
3964         inplace_merge(left.begin(), left.end() - right.size(), left.end());
3965         return (1 << highbit) | self(self, left);
3966     };
3967     vector<array<int, 2>> vals;
3968     for (int i = 0; i < n; ++i) vals.push_back({a[i], i});
3969     sort(begin(vals), end(vals));
3970
3971     int ans = solve(solve, vals);
3972     if (ans == INT_MAX) cout << -1 << '\n';

```

```

3973         else cout << ans << '\n';
3974     }
3975 }
3976
3977 //LS
3978 #include <bits/stdc++.h>
3979 #define ll long long
3980 #define int long long
3981 #define fi first
3982 #define se second
3983 #define mat vector<vector<ll>>
3984 using namespace std;
3985 void db() {cout << endl;}
3986 template <typename T, typename ...U> void db(T a, U ...b) {cout << a << ' ', db(b...);}
3987 #ifdef Cloud
3988 #define file freopen("input.txt", "r", stdin), freopen("output.txt", "w", stdout)
3989 #else
3990 #define file ios::sync_with_stdio(false); cin.tie(0)
3991 #endif
3992 auto SEED = chrono::steady_clock::now().time_since_epoch().count();
3993 mt19937 rng(SEED);
3994 const int N = 2e5 + 5, mod = 998244353, inf = 1ll << 30;
3995
3996 signed main(){
3997     file;
3998     int n, K;
3999     cin >> n >> K;
4000     int dp[n + 1][K + 1], a[n];
4001     for (int &i : a) cin >> i;
4002     for (auto &i : dp) for (int &j : i) j = 0;
4003     int SQ = sqrt(K) * 2 + 5;
4004     for (int i = n - 1; i >= 0; i--){
4005         for (int j = 0; j <= K; j++){
4006             dp[i][j] = dp[i + 1][j];
4007             int sum = 0;
4008             for (int k = i; k < min(n, i + SQ); k++){
4009                 sum += a[k];
4010                 int len = k - i + 1;
4011                 if (len > j) break;
4012                 dp[i][j] = max(dp[i][j], dp[i + 1][j - len] + sum);
4013             }
4014             if (j) dp[i][j] = max(dp[i][j], dp[i][j - 1]);
4015             //db(i, j, dp[i][j]);
4016         }
4017     }
4018     cout << dp[0][K] << '\n';
4019 }
4020
4021 //MIN_UGLY
4022 #include <bits/stdc++.h>
4023 using namespace std;
4024 #define int long long
4025
4026 mt19937_64 RNG(chrono::steady_clock::now().time_since_epoch().count());
4027
4028 const int INF = 1e18, N = 3e5 + 5;
4029 vector<int> g[N];
4030 int dp[20][N];
4031 int depth[N];
4032
4033 void dfs(int node, int par, int dep)
4034 {
4035     depth[node] = dep;
4036     dp[0][node] = par;
4037     for(int to: g[node])
4038     {
4039         if(to == par)
4040         {
4041             continue;

```

```

4042     }
4043     dfs(to, node, dep + 1);
4044 }
4045 }
4046
4047 int dist(int x, int y)
4048 {
4049     if(depth[x] > depth[y])
4050     {
4051         swap(x, y);
4052     }
4053     int diff = depth[y] - depth[x];
4054     int ans = 0;
4055     for(int i = 19; i >= 0; i--)
4056     {
4057         if((1 << i) & diff)
4058         {
4059             y = dp[i][y];
4060             ans += (1 << i);
4061         }
4062     }
4063     if(x == y)
4064     {
4065         return ans;
4066     }
4067     for(int i = 19; i >= 0; i--)
4068     {
4069         if(dp[i][x] != dp[i][y])
4070         {
4071             x = dp[i][x];
4072             y = dp[i][y];
4073             ans += 2 * (1 << i);
4074         }
4075     }
4076     return ans + 2;
4077 }
4078
4079 void Solve()
4080 {
4081     int n, q;
4082     cin >> n >> q;
4083     for(int i = 1; i <= n; i++)
4084     {
4085         g[i].clear();
4086     }
4087     for(int i = 1; i < n; i++)
4088     {
4089         int u, v;
4090         cin >> u >> v;
4091         g[u].push_back(v);
4092         g[v].push_back(u);
4093     }
4094     dfs(1, -1, 0);
4095     for(int i = 1; i < 20; i++)
4096     {
4097         for(int j = 1; j <= n; j++)
4098         {
4099             dp[i][j] = (dp[i - 1][j] == -1) ? dp[i - 1][j] : dp[i - 1][dp[i - 1][j]];
4100         }
4101     }
4102     for(int i = 1; i <= q; i++)
4103     {
4104         // cout << "query number: " << i << "\n";
4105         int k;
4106         cin >> k;
4107         if(k == 1)
4108         {
4109             int u;
4110             cin >> u;

```

```

4111         cout << "0\n";
4112         continue;
4113     }
4114     int u, v;
4115     cin >> u >> v;
4116     int max_dist = dist(u, v);
4117     for(int j = 3; j <= k; j++)
4118     {
4119         int x;
4120         cin >> x;
4121         int diam1 = dist(x, u);
4122         int diam2 = dist(x, v);
4123         if(diam1 > max_dist && diam1 >= diam2)
4124         {
4125             v = x;
4126             max_dist = diam1;
4127         }
4128         else if(diam2 > max_dist && diam2 >= diam1)
4129         {
4130             u = x;
4131             max_dist = diam2;
4132         }
4133     }
4134     cout << (max_dist + 1) / 2 << "\n";
4135 }
4136 }
4137
4138 int32_t main()
4139 {
4140     auto begin = std::chrono::high_resolution_clock::now();
4141     ios_base::sync_with_stdio(0);
4142     cin.tie(0);
4143     int t = 1;
4144     cin >> t;
4145     for(int i = 1; i <= t; i++)
4146     {
4147         //cout << "Case #" << i << ": ";
4148         Solve();
4149     }
4150     auto end = std::chrono::high_resolution_clock::now();
4151     auto elapsed = std::chrono::duration_cast<std::chrono::nanoseconds>(end - begin);
4152     cerr << "Time measured: " << elapsed.count() * 1e-9 << " seconds.\n";
4153     return 0;
4154 }
4155
4156 //FINDINGSUM
4157 #pragma GCC optimization("O3")
4158 #pragma GCC optimize("Ofast,unroll-loops")
4159 #include <bits/stdc++.h>
4160 #include <ext/pb_ds/tree_policy.hpp>
4161 #include <ext/pb_ds/assoc_container.hpp>
4162 using namespace __gnu_pbds;
4163 using namespace std;
4164 #define ll long long
4165 const ll INF_MUL=1e13;
4166 const ll INF_ADD=1e18;
4167 #define pb push_back
4168 #define mp make_pair
4169 #define nline "\n"
4170 #define f first
4171 #define s second
4172 #define pll pair<ll,ll>
4173 #define all(x) x.begin(),x.end()
4174 #define vl vector<ll>
4175 #define vvl vector<vector<ll>>
4176 #define vvvl vector<vector<vector<ll>>>
4177 #ifndef ONLINE_JUDGE
4178 #define debug(x) cerr<<#x<<" "; _print(x); cerr<<nline;
4179 #else

```

```

4180 #define debug(x);
4181 #endif
4182 void _print(int x){cerr<<x;}
4183 void _print(ll x){cerr<<x;}
4184 void _print(char x){cerr<<x;}
4185 void _print(string x){cerr<<x;}
4186 mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());
4187 template<class T,class V> void _print(pair<T,V> p) {cerr<<"{"; _print(p.first);cerr<<" ";
; _print(p.second);cerr<<"}";}
4188 template<class T>void _print(vector<T> v) {cerr<<" [ "; for (T i:v){_print(i);cerr<<" "
; }cerr<<""]";}
4189 template<class T>void _print(set<T> v) {cerr<<" [ "; for (T i:v){_print(i); cerr<<" ";}
cerr<<""]";}
4190 template<class T>void _print(multiset<T> v) {cerr<<" [ "; for (T i:v){_print(i);cerr<<
" ";}cerr<<""]";}
4191 template<class T,class V>void _print(map<T, V> v) {cerr<<" [ "; for(auto i:v) {_print(i
);cerr<<" ";} cerr<<""]";}
4192 typedef tree<ll, null_type, less<ll>, rb_tree_tag, tree_order_statistics_node_update>
ordered_set;
4193 typedef tree<ll, null_type, less_equal<ll>, rb_tree_tag,
tree_order_statistics_node_update> ordered_multiset;
4194 typedef tree<pair<ll,ll>, null_type, less<pair<ll,ll>>, rb_tree_tag,
tree_order_statistics_node_update> ordered_pset;
4195 //-----
-----
4196 const ll MOD=998244353;
4197 const ll MAX=500500;
4198 vector<ll> fact(MAX+2,1),inv_fact(MAX+2,1);
4199 ll binpow(ll a,ll b,ll MOD){
4200     ll ans=1;
4201     a%=MOD;
4202     while(b){
4203         if(b&1)
4204             ans=(ans*a)%MOD;
4205         b/=2;
4206         a=(a*a)%MOD;
4207     }
4208     return ans;
4209 }
4210 ll inverse(ll a,ll MOD){
4211     return binpow(a,MOD-2,MOD);
4212 }
4213 void precompute(ll MOD){
4214     for(ll i=2;i<MAX;i++){
4215         fact[i]=(fact[i-1]*i)%MOD;
4216     }
4217     inv_fact[MAX-1]=inverse(fact[MAX-1],MOD);
4218     for(ll i=MAX-2;i>=0;i--){
4219         inv_fact[i]=(inv_fact[i+1]*(i+1))%MOD;
4220     }
4221 }
4222 ll nCr(ll a,ll b,ll MOD){
4223     if(a==b){
4224         return 1;
4225     }
4226     if((a<0)|| (a<b)|| (b<0))
4227         return 0;
4228     ll denom=(inv_fact[b]*inv_fact[a-b])%MOD;
4229     return (denom*fact[a])%MOD;
4230 }
4231 void solve(){
4232     ll n,m; cin>>n>>m;
4233     ll ans=0;
4234     for(ll i=0;i<=m;i++){
4235         ll mul=(nCr(n+i-1,n-1,MOD)*(211*(m-i)))%MOD;
4236         mul=(mul*(211*(m-i)))%MOD;
4237         for(ll j=0;j<=n;j++){
4238             ll now=(nCr(n,j,MOD)*nCr(m-i-1,j-1,MOD))%MOD;

```

```

4239         now=(now*nCr(m-i+n-j-1,n-j-1,MOD))%MOD;
4240         ans=(ans+now*mul)%MOD;
4241     }
4242 }
4243 cout<<ans<<endl;
4244 return;
4245 }
4246 int main()
4247 {
4248     ios_base::sync_with_stdio(false);
4249     cin.tie(NULL);
4250     #ifndef ONLINE_JUDGE
4251     freopen("input.txt", "r", stdin);
4252     freopen("output.txt", "w", stdout);
4253     freopen("error.txt", "w", stderr);
4254     #endif
4255     ll test_cases=1;
4256     cin>>test_cases;
4257     precompute(MOD);
4258     while(test_cases--){
4259         solve();
4260     }
4261     cout<<fixed<<setprecision(10);
4262     cerr<<"Time:"<<1000*((double)clock())/((double)CLOCKS_PER_SEC<<"ms\n";
4263 }
4264
4265 //SEQGOODNESS
4266 #include <bits/stdc++.h>
4267
4268 using namespace std;
4269
4270 #define int long long int
4271
4272 const int N = 2e5 + 10;
4273 vector<int> fact(N);
4274 vector<int> inv(N);
4275 const int mod = (int)1e9 + 7;
4276
4277 int power(int x, int y, int p){
4278     int res = 1;
4279     x = x % p;
4280     if (x == 0)
4281         return 0;
4282     while (y > 0){
4283         if (y & 1)
4284             res = (res * x) % p;
4285         y = y >> 1;
4286         x = (x * x) % p;
4287     }
4288     return res;
4289 }
4290 void inti() {
4291     fact[0] = 1;
4292     for (int i = 1; i < N; i++){
4293         fact[i] = (fact[i - 1] % mod * i % mod) % mod;
4294     }
4295     for (int i = 0; i < N; i++){
4296         inv[i] = power(fact[i], mod - 2, mod);
4297     }
4298 }
4299 int nCr(int n, int r){
4300     return (fact[n] % mod * inv[n - r] % mod * inv[r] % mod) % mod;
4301 }
4302
4303 int32_t main() {
4304     ios::sync_with_stdio(false);
4305     cin.tie(0);
4306     int tt;
4307     inti();

```

```

4308     cin >> tt;
4309     assert(tt <= 100000);
4310     int sum = 0;
4311     while(tt--) {
4312         int n;
4313         cin >> n;
4314         sum += n;
4315         vector<int> a(n);
4316         for(int i = 0; i < n; i++) {
4317             cin >> a[i];
4318             assert(a[i] <= n);
4319         }
4320         int ans = 0;
4321         sort(a.begin(), a.end());
4322         for(int i = 0; i < n; i++) {
4323             if(a[i] <= i + 1) {
4324                 int right = power(2, n - i - 1, mod);
4325                 int left = nCr(i, a[i] - 1);
4326                 ans += (left * right) % mod;
4327                 ans %= mod;
4328             }
4329         }
4330         assert(ans < mod);
4331         cout << ans << '\n';
4332     }
4333     assert(sum <= 200000);
4334 }
4335
4336 //APP_BAL_SCA
4337 #include <bits/stdc++.h>
4338
4339 using namespace std;
4340
4341 using i64 = int64_t;
4342
4343 int main() {
4344     ios::sync_with_stdio(false);
4345     cin.tie(nullptr);
4346     cout.tie(nullptr);
4347     int t; cin >> t;
4348     assert(1 <= t && t <= 2 * 100000);
4349     while (t--) {
4350         i64 m, n;
4351         cin >> m >> n;
4352         assert(1 <= m && m <= 1e18);
4353         assert(1 <= n && n <= 1e18);
4354         i64 z = m;
4355         if (n > m) {
4356             cout << "NO" << "\n";
4357             continue;
4358         }
4359         while (z % 2 == 0) {
4360             z /= 2;
4361         }
4362         cout << (n % z == 0 ? "YES" : "NO") << "\n";
4363     }
4364     return 0;
4365 }
4366
4367 //CHEFPARTY
4368 #include <bits/stdc++.h>
4369
4370
4371
4372 using namespace std;
4373 #define pb push_back
4374
4375
4376

```



```

4377     const int N = 2e5 + 5;
4378     bool prime[N + 1];
4379     int num_primef[N + 5];
4380
4381
4382     void SieveOfEratosthenes() {
4383         // Create a boolean array "prime[0..n]" and initialize
4384         // all entries it as true. A value in prime[i] will
4385         // finally be false if i is Not a prime, else true.
4386
4387         memset(prime, true, sizeof(prime));
4388
4389         for (int p = 2; p * p <= N; p++) {
4390             // If prime[p] is not changed, then it is a prime
4391             if (prime[p] == true) {
4392                 // Update all multiples of p greater than or
4393                 // equal to the square of it numbers which are
4394                 // multiple of p and are less than p^2 are
4395                 // already been marked.
4396                 for (int i = p * p; i <= N; i += p)
4397                     prime[i] = false;
4398             }
4399         }
4400         prime[1] = false;
4401     }
4402
4403     void precompute() {
4404
4405         num_primef[1] = 0;
4406         for (int i = 2; i <= N; i++) {
4407             if (prime[i]) num_primef[i] = num_primef[i - 1] + 1;
4408             else num_primef[i] = num_primef[i - 1];
4409         }
4410     }
4411
4412
4413     void solve() {
4414
4415         int n;
4416         cin >> n;
4417         int cnt = num_primef[2 * n] - num_primef[n] + 1;
4418         cout << cnt / 2 + cnt % 2 << '\n';
4419
4420         vector< int > s;
4421         for (int i = 2; i <= n; i++) if(prime[i]) s.pb(i);
4422
4423         vector< int > even;
4424         unordered_map< int, int > m;
4425         int sz=s.size();
4426         for (int k=sz-1;k>=0;k--) {
4427
4428             int num = s[k];
4429
4430             int c = 0, i;
4431             for (int j = num; j <= 2 * n; j += num) {
4432                 if (m[j] == 0) c++;
4433             }
4434
4435             if (c % 2 == 0) {
4436
4437                 cout << num << " " << num * 2 << '\n';
4438                 m[num] = 1;
4439                 m[2 * num] = 1;
4440                 i = num * 3;
4441             } else {
4442                 int stop;
4443                 for (int j = num * 3; j <= 2 * n; j += num) {
4444                     if (m[j] == 0) {

```

```

4446         stop = j;
4447         break;
4448     }
4449 }
4450
4451 cout << num << " " << stop << '\n';
4452 m[stop] = 1;
4453 m[num * 2] = 1;
4454 even.pb(num * 2);
4455 i = stop + num;
4456 }
4457 vector < int > v;
4458 for (; i <= 2 * n; i += num) {
4459
4460     if (m[i] == 0) v.pb(i), m[i] = 1;
4461
4462 }
4463 for (int j = 0; j + 1 < v.size(); j += 2)
4464     cout << v[j] << " " << v[j + 1] << '\n';
4465
4466 }
4467
4468 for (int i = 0; i + 1 < even.size(); i += 2) {
4469
4470     cout << even[i] << " " << even[i + 1] << '\n';
4471
4472 }
4473
4474 vector < int > final;
4475 if (even.size() % 2) final.pb(even[even.size() - 1]);
4476 final.pb(1);
4477 for (int i = n + 1; i <= 2 * n; i++)
4478     if (prime[i]) final.pb(i);
4479
4480 for (int i = 0; i + 1 < final.size(); i += 2) {
4481
4482     cout << final[i] << " " << final[i + 1] << '\n';
4483
4484 }
4485
4486 }
4487 signed main() {
4488     ios_base::sync_with_stdio(false);
4489     cin.tie(0);
4490     cout.tie(0);
4491
4492     SieveOfEratosthenes();
4493
4494     precompute();
4495
4496     int t = 1;
4497     cin >> t;
4498     while (t--) solve();
4499     return 0;
4500 }
4501
4502 //PRINTINGBIN
4503 #pragma GCC optimization("O3")
4504 #pragma GCC optimize("Ofast,unroll-loops")
4505 #include <bits/stdc++.h>
4506 #include <ext/pb_ds/tree_policy.hpp>
4507 #include <ext/pb_ds/assoc_container.hpp>
4508 using namespace __gnu_pbds;
4509 using namespace std;
4510 #define ll long long
4511 const ll INF_MUL=1e13;
4512 const ll INF_ADD=1e18;
4513 #define pb push_back
4514 #define mp make_pair

```

```

4515 #define nline "\n"
4516 #define f first
4517 #define s second
4518 #define pll pair<ll,ll>
4519 #define all(x) x.begin(),x.end()
4520 #define vl vector<ll>
4521 #define vvl vector<vector<ll>>
4522 #define vvvl vector<vector<vector<ll>>>
4523 #ifndef ONLINE_JUDGE
4524 #define debug(x) cerr<<#x<<" "; _print(x); cerr<<nline;
4525 #else
4526 #define debug(x);
4527 #endif
4528 void _print(int x){cerr<<x;}
4529 void _print(ll x){cerr<<x;}
4530 void _print(char x){cerr<<x;}
4531 void _print(string x){cerr<<x;}
4532 mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());
4533 template<class T,class V> void _print(pair<T,V> p) {cerr<<"{"; _print(p.first);cerr<<" ";
; _print(p.second);cerr<<"}";}
4534 template<class T>void _print(vector<T> v) {cerr<<" [ "; for (T i:v){_print(i);cerr<<" "
;}cerr<<" ]";}
4535 template<class T>void _print(set<T> v) {cerr<<" [ "; for (T i:v){_print(i); cerr<<" ";}
cerr<<" ]";}
4536 template<class T>void _print(multiset<T> v) {cerr<<" [ "; for (T i:v){_print(i);cerr<<"
";}cerr<<" ]";}
4537 template<class T,class V>void _print(map<T, V> v) {cerr<<" [ "; for(auto i:v) {_print(i
);cerr<<" ";} cerr<<" ]";}
4538 typedef tree<ll, null_type, less<ll>, rb_tree_tag, tree_order_statistics_node_update>
ordered_set;
4539 typedef tree<ll, null_type, less_equal<ll>, rb_tree_tag,
tree_order_statistics_node_update> ordered_multiset;
4540 typedef tree<pair<ll,ll>, null_type, less<pair<ll,ll>>, rb_tree_tag,
tree_order_statistics_node_update> ordered_pset;
4541 //-----
-----
4542 const ll MOD=998244353;
4543 const ll MAX=500500;
4544 vector<ll> fact(MAX+2,1),inv_fact(MAX+2,1);
4545 ll binpow(ll a,ll b,ll MOD){
4546     ll ans=1;
4547     a%=MOD;
4548     while(b){
4549         if(b&1)
4550             ans=(ans*a)%MOD;
4551         b/=2;
4552         a=(a*a)%MOD;
4553     }
4554     return ans;
4555 }
4556 ll inverse(ll a,ll MOD){
4557     return binpow(a,MOD-2,MOD);
4558 }
4559 void precompute(ll MOD){
4560     for(ll i=2;i<MAX;i++){
4561         fact[i]=(fact[i-1]*i)%MOD;
4562     }
4563     inv_fact[MAX-1]=inverse(fact[MAX-1],MOD);
4564     for(ll i=MAX-2;i>=0;i--){
4565         inv_fact[i]=(inv_fact[i+1]*(i+1))%MOD;
4566     }
4567 }
4568 ll nCr(ll a,ll b,ll MOD){
4569     if(a==b){
4570         return 1;
4571     }
4572     if((a<0)|| (a<b)|| (b<0))
4573         return 0;

```

```

4574     ll denom=(inv_fact[b]*inv_fact[a-b])%MOD;
4575     return (denom*fact[a])%MOD;
4576 }
4577 void solve(){
4578     ll n; cin>>n;
4579     for(ll i=1;i<=n;i++){
4580         ll x; cin>>x;
4581         x=1-x;
4582         cout<<x<<" \n"[i==n];
4583     }
4584     return;
4585 }
4586 int main()
4587 {
4588     ios_base::sync_with_stdio(false);
4589     cin.tie(NULL);
4590     #ifndef ONLINE_JUDGE
4591     freopen("input.txt", "r", stdin);
4592     freopen("output.txt", "w", stdout);
4593     freopen("error.txt", "w", stderr);
4594     #endif
4595     ll test_cases=1;
4596     cin>>test_cases;
4597     precompute(MOD);
4598     while(test_cases--){
4599         solve();
4600     }
4601     cout<<fixed<<setprecision(10);
4602     cerr<<"Time:"<<1000*((double)clock())/(double)CLOCKS_PER_SEC<<"ms\n";
4603 }
4604
4605 //TWOAVG
4606 #include <bits/stdc++.h>
4607
4608 using namespace std;
4609
4610 void test_case(){
4611     int n, m, k;
4612     cin >> n >> m >> k;
4613
4614     vector<int> a(n), b(m);
4615     for (int i = 0; i < n; i++) cin >> a[i];
4616     for (int i = 0; i < m; i++) cin >> b[i];
4617
4618     if (k == 1){
4619         cout << -1 << endl;
4620         return;
4621     }
4622
4623     long long sumA = accumulate(a.begin(), a.end(), 0LL);
4624     long long sumB = accumulate(b.begin(), b.end(), 0LL);
4625
4626     int ans = n + m + 2;
4627     int Y = n + m + 2;
4628     for (int X = 0; X <= n + m + 2; X++){
4629         while (Y >= 0 && (sumA + (long long) X * k) * (m + Y) > (sumB + Y) * (n + X)) Y
            --;
4630         Y++;
4631         ans = min(ans, X + Y);
4632     }
4633
4634     cout << ans << endl;
4635 }
4636
4637 int main(){
4638     ios_base::sync_with_stdio(false);
4639
4640     #ifdef LOCAL
4641     freopen("input.txt", "r", stdin);

```

```

4642     freopen("output.txt", "w", stdout);
4643 #endif
4644
4645     int T;
4646     cin >> T;
4647
4648     while (T--){
4649         test_case();
4650     }
4651
4652     return 0;
4653 }
4654
4655 //EVIL_INF
4656 #include <map>
4657 #include <set>
4658 #include <cmath>
4659 #include <ctime>
4660 #include <queue>
4661 #include <stack>
4662 #include <cstdio>
4663 #include <cstdlib>
4664 #include <vector>
4665 #include <cstring>
4666 #include <algorithm>
4667 #include <iostream>
4668 using namespace std;
4669 typedef double db;
4670 typedef long long ll;
4671 typedef unsigned long long ull;
4672 const int N=1000010;
4673 const int LOGN=28;
4674 const ll TMD=0;
4675 const ll INF=2147483647;
4676 int T,n;
4677 int steal[N],dp[N];
4678
4679 struct Data
4680 {
4681     int t,num,ty;
4682
4683     Data() {}
4684
4685     Data(int t,int num,int ty):t(t),num(num),ty(ty) {}
4686
4687     friend bool operator<(Data x,Data y)
4688     {
4689         return x.t<y.t;
4690     }
4691 }d[N];
4692
4693 void init()
4694 {
4695     scanf("%d",&n);
4696     for(int i=1;i<=n;i++)
4697     {
4698         int t;
4699         scanf("%d",&t);
4700         d[i]=Data(t,i,0);
4701     }
4702     for(int i=1;i<=n;i++)
4703     {
4704         int t;
4705         scanf("%d",&t);
4706         d[n+i]=Data(t,i,1);
4707     }
4708     sort(d+1,d+2*n+1);
4709 }
4710

```

```

4711 void solve()
4712 {
4713     int ans=0;
4714     set<int> S;
4715     for(int i=1;i<=n;i++) dp[i]=steal[i]=0;
4716     for(int i=1;i<=n*2;i++)
4717     {
4718         if(d[i].ty)
4719         {
4720             S.erase(d[i].num);
4721             if(!S.empty()) steal[d[i].num]=*S.begin();
4722         }
4723         else S.insert(d[i].num);
4724     }
4725     for(int i=n*2;i;i--)
4726     {
4727         if(d[i].ty&&steal[d[i].num])
4728         {
4729             dp[d[i].num]=dp[steal[d[i].num]]+1;
4730             ans=max(ans,dp[d[i].num]);
4731         }
4732     }
4733     printf("%d\n",ans);
4734 }
4735
4736 int main()
4737 {
4738     scanf("%d",&T);
4739     while(T--)
4740     {
4741         init();
4742         solve();
4743     }
4744
4745     return 0;
4746 }
4747
4748 //REMSUBARR
4749 #ifdef WTSH
4750     #include <wtsh.h>
4751 #else
4752     #include <bits/stdc++.h>
4753     using namespace std;
4754     #define dbg(...)
4755 #endif
4756
4757 #define int long long
4758 #define endl "\n"
4759 #define sz(w) (int)(w.size())
4760 using pii = pair<int, int>;
4761
4762 const long long INF = 1e18;
4763
4764 const int N = 1e6 + 5;
4765
4766 // ----- Input Checker Start -----
4767
4768 long long readInt(long long l, long long r, char endd)
4769 {
4770     long long x = 0;
4771     int cnt = 0, fi = -1;
4772     bool is_neg = false;
4773     while(true)
4774     {
4775         char g = getchar();
4776         if(g == '-')
4777         {
4778             assert(fi == -1);
4779             is_neg = true;

```

```

4780         continue;
4781     }
4782     if('0' <= g && g <= '9')
4783     {
4784         x *= 10;
4785         x += g - '0';
4786         if(cnt == 0)
4787             fi = g - '0';
4788         cnt++;
4789         assert(fi != 0 || cnt == 1);
4790         assert(fi != 0 || is_neg == false);
4791         assert(!(cnt > 19 || (cnt == 19 && fi > 1)));
4792     }
4793     else if(g == endd)
4794     {
4795         if(is_neg)
4796             x = -x;
4797         if(!(l <= x && x <= r))
4798         {
4799             cerr << "L: " << l << ", R: " << r << ", Value Found: " << x << '\n';
4800             assert(false);
4801         }
4802         return x;
4803     }
4804     else
4805     {
4806         assert(false);
4807     }
4808 }
4809
4810 string readString(int l, int r, char endd)
4811 {
4812     string ret = "";
4813     int cnt = 0;
4814     while(true)
4815     {
4816         char g = getchar();
4817         assert(g != -1);
4818         if(g == endd)
4819             break;
4820         cnt++;
4821         ret += g;
4822     }
4823     assert(l <= cnt && cnt <= r);
4824     return ret;
4825 }
4826
4827 long long readIntSp(long long l, long long r) { return readInt(l, r, ' '); }
4828 long long readIntLn(long long l, long long r) { return readInt(l, r, '\n'); }
4829 string readStringSp(int l, int r) { return readString(l, r, ' '); }
4830 string readStringLn(int l, int r) { return readString(l, r, '\n'); }
4831 void readEOF() { assert(getchar() == EOF); }
4832
4833 vector<int> readVectorInt(int n, long long l, long long r)
4834 {
4835     vector<int> a(n);
4836     for(int i = 0; i < n - 1; i++)
4837         a[i] = readIntSp(l, r);
4838     a[n - 1] = readIntLn(l, r);
4839     return a;
4840 }
4841
4842 // ----- Input Checker End -----
4843
4844 int sumN = 0;
4845
4846 void solve()
4847 {

```

```

4849     int n = readIntLn(2, 1e5);
4850     sumN += n;
4851     vector<int> a = readVectorInt(n, 1, n);
4852     assert(set<int>(a.begin(), a.end()).size() == n);
4853     vector<int> idx(n + 1);
4854     for(int i = 0; i < n; i++)
4855         idx[a[i]] = i;
4856     int L = N + 1, R = 0, ans = 0;
4857     for(int i = n; i > 1; i--)
4858     {
4859         L = min(L, idx[i]);
4860         R = max(R, idx[i]);
4861         if(R - L + 1 == n - i + 1)
4862             ans = R - L + 1;
4863     }
4864     cout << ans << endl;
4865 }
4866
4867 int32_t main()
4868 {
4869     ios::sync_with_stdio(0);
4870     cin.tie(0);
4871     int T = readIntLn(1, 1e5);
4872     for(int tc = 1; tc <= T; tc++)
4873     {
4874         // cout << "Case #" << tc << ": ";
4875         solve();
4876     }
4877     readEOF();
4878     assert(sumN <= 5e5);
4879     cerr << sumN << endl;
4880     return 0;
4881 }
4882
4883 //DDMMORMMDD
4884 #include <iostream>
4885
4886 int main() {
4887     int tests;
4888     std::cin >> tests;
4889     while (tests--) {
4890         std::string s;
4891         std::cin >> s;
4892         int x = (s[0] - '0') * 10 + (s[1] - '0');
4893         int y = (s[3] - '0') * 10 + (s[4] - '0');
4894         if (x <= 12 and y <= 12)
4895             std::cout << "BOTH" << '\n';
4896         else if (y <= 12)
4897             std::cout << "DD/MM/YYYY" << '\n';
4898         else
4899             std::cout << "MM/DD/YYYY" << '\n';
4900     }
4901 }
4902
4903 //RESTORECOM
4904 #include<bits/stdc++.h>
4905 using namespace std;
4906 using ll=long long;
4907
4908 #ifdef ANI
4909 #include "D:/DUSTBIN/local_inc.h"
4910 #else
4911 #define dbg(...) 0
4912 #endif
4913
4914 class dsu{
4915 public:
4916     int N;
4917     vector<int> par,size;

```



```

4918     dsu(int N) {
4919         this->N=N; par=size=vector<int>(N,1);
4920         for(int i=0;i<N;i++)
4921             par[i]=i;
4922     }
4923     int find(int u) {
4924         return par[u]==u?u:par[u]=find(par[u]);
4925     }
4926     int join(int u,int v) {
4927         u=find(u),v=find(v);
4928         if(u==v) return 0;
4929         if(size[u]<size[v]) swap(u,v);
4930         par[v]=u; size[u]+=size[v]; return 1;
4931     }
4932 };
4933
4934 int main() {
4935     ios_base::sync_with_stdio(false);cin.tie(NULL);cout.tie(NULL);
4936     int t;
4937     cin>>t;
4938     while(t--) {
4939         int n,ect=0; cin>>n;
4940         vector<int> u;
4941         vector<vector<int>> frq(n);
4942         for(int i=0;i<n;i++) {
4943             int k; cin>>k;
4944             ect+=k;
4945             while(k--) {
4946                 int x; cin>>x;
4947                 u.push_back(x);
4948                 frq[i].push_back(x);
4949             }
4950         }
4951         sort(u.begin(),u.end());
4952         u.resize(unique(u.begin(),u.end())-u.begin());
4953         int m=u.size();
4954
4955         if(ect<n+m-1) {
4956             cout<<-1<<"\n";
4957             continue;
4958         }
4959         dsu D(n+m);
4960
4961         for(int i=0;i<n;i++) {
4962             for(int f:frq[i]) {
4963                 int k=lower_bound(u.begin(),u.end(),f)-u.begin()+n;
4964                 D.join(i,k);
4965             }
4966         }
4967
4968         int ans=-1;
4969         for(int i=0;i<n;i++) {
4970             ans+=(D.find(i)==i);
4971         }
4972         cout<<ans<<"\n";
4973     }
4974 }
4975
4976 //TREEQR
4977 // library link: https://github.com/manan-grover/My-CP-Library/blob/main/library.cpp
4978 #include <bits/stdc++.h>
4979 #include <ext/pb_ds/assoc_container.hpp>
4980 #include <ext/pb_ds/tree_policy.hpp>
4981 using namespace std;
4982 using namespace __gnu_pbds;
4983 #define asc(i,a,n) for(I i=a;i<n;i++)
4984 #define dsc(i,a,n) for(I i=n-1;i>=a;i--)
4985 #define forw(it,x) for(A it=(x).begin();it!=(x).end();it++)
4986 #define bacw(it,x) for(A it=(x).rbegin();it!=(x).rend();it++)

```

```

4987 #define pb push_back
4988 #define mp make_pair
4989 #define fi first
4990 #define se second
4991 #define lb(x) lower_bound(x)
4992 #define ub(x) upper_bound(x)
4993 #define fbo(x) find_by_order(x)
4994 #define ook(x) order_of_key(x)
4995 #define all(x) (x).begin(), (x).end()
4996 #define sz(x) (I)((x).size())
4997 #define clr(x) (x).clear()
4998 #define U unsigned
4999 #define I int
5000 #define S string
5001 #define C char
5002 #define D long double
5003 #define A auto
5004 #define B bool
5005 #define CM(x) complex<x>
5006 #define V(x) vector<x>
5007 #define P(x,y) pair<x,y>
5008 #define OS(x) set<x>
5009 #define US(x) unordered_set<x>
5010 #define OMS(x) multiset<x>
5011 #define UMS(x) unordered_multiset<x>
5012 #define OM(x,y) map<x,y>
5013 #define UM(x,y) unordered_map<x,y>
5014 #define OMM(x,y) multimap<x,y>
5015 #define UMM(x,y) unordered_multimap<x,y>
5016 #define BS(x) bitset<x>
5017 #define L(x) list<x>
5018 #define Q(x) queue<x>
5019 #define PBS(x) tree<x,null_type,less<I>,rb_tree_tag,tree_order_statistics_node_update>
5020 #define PBM(x,y) tree<x,y,less<I>,rb_tree_tag,tree_order_statistics_node_update>
5021 #define pi (D)acos(-1)
5022 #define md 1000000007
5023 #define rnd randGen(rng)
5024 inline void comp(P(P(I,I),I) &vv,P(P(I,I),I) v[2],I y){
5025     if(vv.fi.fi>v[0].fi.fi){
5026         v[1]=v[0];
5027         v[0]=vv;
5028         v[0].se=y;
5029     }else if(vv.fi.fi>v[1].fi.fi){
5030         v[1]=vv;
5031         v[1].se=y;
5032     }
5033 }
5034 void dfs0(I x,I pr,V(P(I,I)) tr[],P(P(I,I),I) v[][2],I temp){
5035     P(P(I,I),I) vv;
5036     asc(i,0,sz(tr[x])){
5037         I y=tr[x][i].fi;
5038         I w=tr[x][i].se;
5039         if(y!=pr){
5040             dfs0(y,x,tr,v,temp);
5041             if(w&temp){
5042                 if(v[y][0].fi.fi!=-1){
5043                     vv=v[y][0];
5044                     vv.fi.fi++;
5045                 }else{
5046                     vv={{1,y},y};
5047                 }
5048             }else{
5049                 vv={{0,x},y};
5050             }
5051             comp(vv,v[x],y);
5052         }
5053     }
5054 }
5055 void dfs1(I x,I pr,V(P(I,I)) tr[],P(P(I,I),I) v[][2],I temp){

```

```

5056     P(P(I,I),I) vv;
5057     if(x==1){
5058         vv={{0,1},0};
5059         comp(vv,v[1],0);
5060     }
5061     asc(i,0,sz(tr[x])){
5062         I y=tr[x][i].fi;
5063         I w=tr[x][i].se;
5064         if(y!=pr){
5065             if(w&temp){
5066                 if(v[x][0].se!=y){
5067                     vv=v[x][0];
5068                 }else{
5069                     vv=v[x][1];
5070                 }
5071                 vv.fi.fi++;
5072             }else{
5073                 vv={{0,y},x};
5074             }
5075             comp(vv,v[y],x);
5076             dfs1(y,x,tr,v,temp);
5077         }
5078     }
5079 }
5080 void cal(V(I) dp[],V(P(I,I)) tr[],I n,I temp){
5081     P(P(I,I),I) v[n+1][2]; //{len, dest, dir};
5082     asc(i,1,n+1){
5083         v[i][0]=v[i][1]={{-1,-1},-1};
5084     }
5085     dfs0(1,0,tr,v,temp);
5086     dfs1(1,0,tr,v,temp);
5087     asc(i,1,n+1){
5088         /*if(sz(v[i][0])){
5089             cout<<i<<" "<<v[i][0][0]<<"\n";
5090         }*/
5091         if(v[i][1].fi.fi==1){
5092             dp[i]={v[i][0].fi.fi,i,v[i][0].fi.se}; //{len, start, end};
5093         }else{
5094             dp[i]={v[i][0].fi.fi+v[i][1].fi.fi,v[i][0].fi.se,v[i][1].fi.se};
5095         }
5096     }
5097 }
5098 int main(){
5099     mt19937_64 rng(chrono::steady_clock::now().time_since_epoch().count());
5100     uniform_int_distribution<I> randGen;
5101     ios_base::sync_with_stdio(false);cin.tie(NULL);cout.tie(NULL);
5102     #ifndef ONLINE_JUDGE
5103     freopen("input.txt", "r", stdin);
5104     freopen("output.txt", "w", stdout);
5105     #endif
5106     I t;
5107     cin>>t;
5108     while(t--){
5109         I n;
5110         cin>>n;
5111         V(P(I,I)) tr[n+1];
5112         asc(i,0,n-1){
5113             I u,v,w;
5114             cin>>u>>v>>w;
5115             tr[u].pb({v,w});
5116             tr[v].pb({u,w});
5117         }
5118         V(I) dp[30][n+1];
5119         I temp=1;
5120         asc(i,0,30){
5121             cal(dp[i],tr,n,temp);
5122             temp*=2;
5123         }
5124         I q;

```

```

5125     cin>>q;
5126     while(q--){
5127         I x;
5128         cin>>x;
5129         I res=-1;
5130         I u,v;
5131         asc(i,0,30){
5132             if(dp[i][x][0]>res){
5133                 res=dp[i][x][0];
5134                 u=dp[i][x][1];
5135                 v=dp[i][x][2];
5136             }
5137         }
5138         //cout<<res<<"\n";
5139         if(res){
5140             cout<<u<<" "<<v<<"\n";
5141         }else{
5142             cout<<-1<<" "<<-1<<"\n";
5143         }
5144     }
5145 }
5146 return 0;
5147 }
5148
5149 //ROIDS
5150 #include <bits/stdc++.h>
5151
5152 #define el '\n'
5153
5154 typedef long long ll;
5155 typedef long double ld;
5156
5157 #define Beevo ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0);
5158
5159 using namespace std;
5160
5161 const int N = 1e5 + 5, M = 1e9 + 7;
5162
5163 int fact[N], freq[N];
5164
5165 int mul(int a, int b) {
5166     return 1LL * a * b % M;
5167 }
5168
5169 void pre() {
5170     fact[0] = 1;
5171
5172     for (int i = 1; i < N; i++)
5173         fact[i] = mul(fact[i - 1], i);
5174 }
5175
5176 int modPow(int b, int p) {
5177     if (p == 0)
5178         return 1;
5179
5180     int x = modPow(b, p / 2);
5181
5182     return p % 2 == 0 ? mul(x, x) : mul(b, mul(x, x));
5183 }
5184
5185 int modInvFer(int n) {
5186     return modPow(n, M - 2);
5187 }
5188
5189 void testCase() {
5190     pre();
5191
5192     int n;
5193     cin >> n;

```

```

5194     assert(n >= 1 && n <= 1e5);
5195
5196     int x, y;
5197     for (int i = 0; i < n; i++)
5198         cin >> x >> y, freq[x]++, assert(min(x, y) >= 1 && max(x, y) <= 1e5);
5200
5201     int ways = 1;
5202     for (int i = 0; i < N; i++)
5203         ways = mul(ways, fact[freq[i]]);
5204
5205     cout << mul(ways, modInvFer(fact[n]));
5206 }
5207
5208 signed main() {
5209     Beevo
5210
5211     int t = 1;
5212     //     cin >> t;
5213
5214     while (t--)
5215         testCase();
5216 }
5217
5218 //MINBITS
5219 #include <bits/stdc++.h>
5220 #include <iostream>
5221 #include <cassert>
5222 #define int long long
5223 #define rep(i,a,n) for(int i = a; i < n; i++)
5224 #define repr(i,n) for(int i = n-1; i >= 0; i--)
5225 #define ff first
5226 #define ss second
5227 #define pb push_back
5228 #define ub upper_bound
5229 #define lb lower_bound
5230 #define sz(x) (x).size()
5231 #define all(x) x.begin(), x.end()
5232 #define mems(x,a) memset((x), a, sizeof(x))
5233 const char nl = '\n';
5234 const int INF = LONG_MAX;
5235 using namespace std;
5236 using namespace __gnu_cxx;
5237 // READ TEMPLATE
5238 void read(){}
5239 void read(unsigned int&a){cin>>a;}
5240 void read(int& a){cin>>a;}
5241 void read(double& a){cin>>a;}
5242 void read(float& a){cin>>a;}
5243 void read(string& a){cin>>a;}
5244 template<typename x,typename y>void read(pair<x,y>& a){ read(a.first),read(a.second);}
5245 template<typename x>void read(x& a){for(auto &i : a) read(i);}
5246 template<typename x,typename... y>void read(x& a,y&... b){read(a);read(b...);}
5247 // DEBUG TEMPLATE
5248 void _print(char i){ cout<<i;}
5249 void _print(string i){ cout<<i;}
5250 void _print(float i){ cout<<i;}
5251 void _print(int i){ cout<<i;}
5252 void _print(double i){ cout<<i;}
5253 void _print(bool i){ cout<<i;}
5254 void _print(long double i){ cout<<i;}
5255 void _print(){cout<<"\n";}
5256 template<typename x,typename y> void _print(pair<x,y>&t){cout<<"{ ";_print(t.first);cout
<<" , ";_print(t.second);cout<<" },";}
5257 template<typename x> void _print(x &t){ cout<<"{ "; for(int i = 0;i < (int)t.size();i
++){ _print(t[i]); if(i < (int) t.size() - 1) cout<<" , "; } cout<<" }"; }
5258 template<typename x,typename... y> void _print(x a,y... b){_print(a);if(sizeof...(b))
cout<<" , ";_print(b...);}
5259 #define dbg(x...) cout<<"DEBUG : "<<#x<<" => ";_print(x);cout<<"\n";

```

```

5260
5261 vector<int> prime(int a, int b){
5262 vector<bool> v(b+1, 1);
5263 vector<int> ans;
5264 rep(i,2,b+1){
5265 if(v[i]){
5266 for(int j = 2*i; j <= b; j += i){
5267 v[j] = 0;
5268 }
5269 if(i >= a) {ans.pb(i);};
5270 }
5271 }
5272 return ans;
5273 }
5274
5275
5276 bool subtract(string &a, string &b, string &c) {
5277
5278 int n = a.size();
5279
5280 /* Going from lsb to msb so we can ask for borrow */
5281 for(int i = 0; i < n; i++){
5282 int x = b[i]-'0';
5283 int y = c[i]-'0';
5284 int z = x-y;
5285 if(z < 0){
5286 int j = i;
5287 while(j < n and b[j] == '0') j++;
5288 if(j != n){
5289 z = 1;
5290 b[j--] = '0';
5291 while(j >= i){
5292 b[j--] = '1';
5293 }
5294 }
5295 else return false;
5296 }
5297
5298 if(a[i] != '0'+z) return false;
5299 }
5300 return true;
5301 }
5302 }
5303
5304 bool test_case = 1;
5305 void solve() {
5306 int n;
5307 cin >> n;
5308 string a;
5309 cin >> a;
5310 reverse(a.begin(), a.end());
5311 string b="",c="";
5312 int cnt=0;
5313 for(int i=0;i<n;i++)
5314 {
5315 if(a[i]=='1')
5316 cnt++;
5317 else
5318 {
5319 if(cnt>=1)
5320 {
5321 c+='1';
5322 for(int j=0;j<cnt;j++)
5323 {
5324 b+='0';
5325 c+='0';
5326 }
5327 b+='1';
5328 }

```

```

5329         else
5330         {
5331             b+='0';
5332             c+='0';
5333         }
5334         cnt=0;
5335     }
5336 }
5337 while(b.length()<n)
5338 {
5339     b+='1';
5340     c+='0';
5341 }
5342 for(int i=0;i<n-1;i++)
5343 {
5344     if(b[i]=='1' && c[i+1]=='1')
5345     {
5346         b[i]='0';
5347         c[i]='1';
5348         c[i+1]='0';
5349     }
5350     else if(b[i+1]=='1' && c[i]=='1')
5351     {
5352         b[i]='1';
5353         c[i]='0';
5354         b[i+1]='0';
5355     }
5356 }
5357 reverse(b.begin(), b.end());
5358 reverse(c.begin(), c.end());
5359
5360 cout << b << '\n' << c << '\n';
5361 }
5362
5363 signed main() {
5364
5365     ios_base::sync_with_stdio(false);
5366     cin.tie(NULL);
5367
5368     #ifndef ONLINE_JUDGE
5369     freopen("input.txt" , "r" , stdin) ;
5370     freopen("output.txt" , "w" , stdout) ;
5371     freopen("error.txt" , "w" , stderr) ;
5372     #endif
5373
5374     int T = 1;
5375     if(test_case) cin>>T;
5376
5377
5378     while( T-- ){
5379         solve();
5380     }
5381
5382
5383     return 0;
5384 }
5385
5386
5387 //MIN_OR_ST
5388 #include <map>
5389 #include <set>
5390 #include <cmath>
5391 #include <ctime>
5392 #include <queue>
5393 #include <stack>
5394 #include <cstdio>
5395 #include <cstdlib>
5396 #include <vector>
5397 #include <cstring>

```

```

5398 #include <algorithm>
5399 #include <iostream>
5400 using namespace std;
5401 typedef double db;
5402 typedef long long ll;
5403 typedef unsigned long long ull;
5404 const int N=200010;
5405 const int LOGN=28;
5406 const ll TMD=0;
5407 const ll INF=2147483647;
5408 int T,n,m,q;
5409 int ty[N],tag[N],ans[N];
5410
5411 struct Edge
5412 {
5413     int u,v,w;
5414
5415     Edge() {}
5416
5417     Edge(int u,int v,int w):u(u),v(v),w(w) {}
5418 };
5419 vector<Edge> E,Q;
5420
5421 int f[N];
5422 int find(int x)
5423 {
5424     return x==f[x]?x:f[x]=find(f[x]);
5425 }
5426
5427 void uni(int x,int y)
5428 {
5429     f[find(x)]=find(y);
5430 }
5431
5432 int cal_component(vector<Edge> &e)
5433 {
5434     int cnt=0;
5435     for(int i=1;i<=n;i++) tag[i]=0;
5436     for(int i=1;i<=n;i++) f[i]=i;
5437     for(int i=0;i<e.size();i++) uni(e[i].u,e[i].v);
5438     for(int i=1;i<=n;i++)
5439         if(!tag[find(i)]) tag[find(i)]=1,cnt++;
5440     return cnt;
5441 }
5442
5443 int cal_answer(vector<Edge> e)
5444 {
5445     int ans=0;
5446     for(int i=29;i>=0;i--)
5447     {
5448         vector<Edge> tmp;
5449         for(int j=0;j<e.size();j++)
5450             if(!(e[j].w&(1<<i))) tmp.push_back(e[j]);
5451         if(cal_component(tmp)!=1) ans^=(1<<i);
5452         else e=tmp;
5453     }
5454     return ans;
5455 }
5456
5457 int main()
5458 {
5459     scanf("%d%d%d",&n,&m,&q);
5460     for(int i=1;i<=m;i++)
5461     {
5462         int u,v,w;
5463         scanf("%d%d%d",&u,&v,&w);
5464         E.push_back(Edge(u,v,w));
5465     }
5466     for(int i=1;i<=q;i++)

```



```

5467     {
5468         int u,v;
5469         scanf("%d%d",&u,&v);
5470         Q.push_back(Edge(u,v,0));
5471     }
5472     for(int i=29;i>=0;i--)
5473     {
5474         vector<Edge> tmp;
5475         for(int j=0;j<E.size();j++)
5476             if(!(E[j].w&(1<<i))) tmp.push_back(E[j]);
5477         if(cal_component(tmp)==1) E=tmp;
5478     }
5479     for(int i=29;i>=0;i--)
5480     {
5481         int c;
5482         vector<Edge> tmp;
5483         for(int j=0;j<E.size();j++)
5484             if(!(E[j].w&(1<<i))) tmp.push_back(E[j]);
5485         c=cal_component(tmp);
5486         if(c==1||c>2) continue;
5487         else
5488         {
5489             for(int j=0;j<Q.size();j++)
5490                 if((!ty[j])&&find(Q[j].u)!=find(Q[j].v)) ty[j]=i+1;
5491         }
5492     }
5493     ans[0]=cal_answer(E);
5494     for(int i=1;i<=30;i++) ans[i]=-1;
5495     for(int i=0;i<Q.size();i++)
5496     {
5497         if(ans[ty[i]]==-1)
5498         {
5499             E.push_back(Q[i]);
5500             ans[ty[i]]=cal_answer(E);
5501             E.pop_back();
5502         }
5503         printf("%d\n",ans[ty[i]]);
5504     }
5505
5506     return 0;
5507 }
5508
5509 //GOOD_OPR
5510 #include <map>
5511 #include <set>
5512 #include <cmath>
5513 #include <ctime>
5514 #include <queue>
5515 #include <stack>
5516 #include <cstdio>
5517 #include <cstdlib>
5518 #include <vector>
5519 #include <cstring>
5520 #include <algorithm>
5521 #include <iostream>
5522 using namespace std;
5523 typedef double db;
5524 typedef long long ll;
5525 typedef unsigned long long ull;
5526 const int N=200010;
5527 const int LOGN=20;
5528 const ll TMD=998244353;
5529 const ll INF=2147483647;
5530 int T,n,m;
5531 int l[N],r[N];
5532 ll f2[N],f3[N];
5533 ll dp[N][LOGN][2];
5534
5535 ll pw(ll x,ll p)

```

```

5536 {
5537     if(!p) return 1;
5538     ll y=pw(x,p>>1);
5539     y=(y*y)%TMD;
5540     if(p&1) y=(y*(x%TMD))%TMD;
5541     return y;
5542 }
5543
5544 ll inv(ll x)
5545 {
5546     return pw(x,TMD-2);
5547 }
5548
5549 ll sum(int t,ll x)
5550 {
5551     if(x==0) return 0;
5552     if(t==1) return x*(x+1)%TMD*inv(2)%TMD;
5553     else if(t==2) return x*(x+1)%TMD*(2*x+1)%TMD*inv(6)%TMD;
5554     else return pw(sum(1,x),2);
5555 }
5556
5557 ll sum(int t,int l,int r)
5558 {
5559     return (sum(t,r)-sum(t,l-1)+TMD)%TMD;
5560 }
5561
5562 void cal_f()
5563 {
5564     for(int i=1;i<=n;i++)
5565     {
5566         ll s1=sum(1,l[i],r[i]),s2=sum(2,l[i],r[i]),s3=sum(3,l[i],r[i]);
5567         f2[i]=(pw(s1,2)-s2+TMD)*inv(2)%TMD;
5568         if(r[i]-l[i]+1>2) f3[i]=(pw(s1,3)-3*s1*s2%TMD+2*s3+TMD*TMD)%TMD*inv(6)%TMD;
5569         else f3[i]=0;
5570     }
5571 }
5572
5573 int main()
5574 {
5575     scanf("%d",&T);
5576     while(T--)
5577     {
5578         scanf("%d",&n);
5579         for(int i=1;i<=n;i++) scanf("%d%d",&l[i],&r[i]);
5580         cal_f();
5581         m=(int)log2(n);
5582         for(int i=1;i<=n;i++)
5583         {
5584             dp[i-1][0][0]=1;
5585             for(int j=1;j<=m;j++)
5586             {
5587                 for(int k=0;k<=1;k++)
5588                 {
5589                     dp[i][j][k]=dp[i-1][j][k];
5590                     dp[i][j][k]=(dp[i][j][k]+dp[i-1][j-1][k]*f2[i])%TMD;
5591                     if(k) dp[i][j][k]=(dp[i][j][k]+dp[i-1][j-1][0]*f3[i])%TMD;
5592                 }
5593             }
5594         }
5595         if((1<=(m-1))*3<=n) printf("%d\n", (dp[n][m][0]+dp[n][m][1])%TMD);
5596         else printf("%d\n",dp[n][m][0]);
5597     }
5598
5599     return 0;
5600 }
5601
5602 //COUNT_PERM
5603 #include <bits/stdc++.h>
5604

```

```

5605 using namespace std;
5606
5607 using ll = long long;
5608 const ll MOD = 998244353;
5609
5610 int main() {
5611     cin.tie(0)->sync_with_stdio(0);
5612
5613     int t; cin >> t;
5614     while (t--) {
5615         int n, k; cin >> n >> k;
5616
5617         vector<bool> skip(n + 1);
5618         skip[n] = true;
5619         for (int i = 0; i < k; i++) {
5620             int a; cin >> a;
5621             skip[n - a] = true;
5622         }
5623
5624         ll ans = 1;
5625         for (int i = 1; i <= n; i++) {
5626             if (!skip[i]) ans = (ans * i) % MOD;
5627         }
5628         cout << ans << "\n";
5629     }
5630 }
5631
5632 //ONEORALL
5633 //Utkarsh.25dec
5634 #include <iostream>
5635 #include <cstdio>
5636 #include <cstdlib>
5637 #include <algorithm>
5638 #include <cmath>
5639 #include <vector>
5640 #include <set>
5641 #include <map>
5642 #include <unordered_set>
5643 #include <unordered_map>
5644 #include <queue>
5645 #include <ctime>
5646 #include <cassert>
5647 #include <complex>
5648 #include <string>
5649 #include <cstring>
5650 #include <chrono>
5651 #include <random>
5652 #include <bitset>
5653 #include <array>
5654 #define ll long long int
5655 #define pb push_back
5656 #define mp make_pair
5657 #define mod 1000000007
5658 #define vl vector<ll>
5659 #define all(c) (c).begin(), (c).end()
5660 using namespace std;
5661 ll power(ll a, ll b) {ll res=1; a%=mod; assert(b>=0); for(; b>=1; b--){if(b&1) res=res*a%mod; a=a*a%mod;} return res;}
5662 ll modInverse(ll a){return power(a,mod-2);}
5663 const int N=500023;
5664 bool vis[N];
5665 vector<int> adj[N];
5666 long long readInt(long long l, long long r, char endd){
5667     long long x=0;
5668     int cnt=0;
5669     int fi=-1;
5670     bool is_neg=false;
5671     while(true){
5672         char g=getchar();

```

```

5673         if(g=='-'){
5674             assert(fi==-1);
5675             is_neg=true;
5676             continue;
5677         }
5678         if('0'<=g && g<='9'){
5679             x*=10;
5680             x+=g-'0';
5681             if(cnt==0){
5682                 fi=g-'0';
5683             }
5684             cnt++;
5685             assert(fi!=0 || cnt==1);
5686             assert(fi!=0 || is_neg==false);
5687
5688             assert(!(cnt>19 || ( cnt==19 && fi>1) ));
5689         } else if(g==endd){
5690             if(is_neg){
5691                 x= -x;
5692             }
5693
5694             if(!(l <= x && x <= r))
5695             {
5696                 cerr << l << ' ' << r << ' ' << x << '\n';
5697                 assert(1 == 0);
5698             }
5699
5700             return x;
5701         } else {
5702             assert(false);
5703         }
5704     }
5705 }
5706 string readString(int l,int r,char endd){
5707     string ret="";
5708     int cnt=0;
5709     while(true){
5710         char g=getchar();
5711         assert(g!=-1);
5712         if(g==endd){
5713             break;
5714         }
5715         cnt++;
5716         ret+=g;
5717     }
5718     assert(l<=cnt && cnt<=r);
5719     return ret;
5720 }
5721 long long readIntSp(long long l,long long r){
5722     return readInt(l,r,' ');
5723 }
5724 long long readIntLn(long long l,long long r){
5725     return readInt(l,r,'\n');
5726 }
5727 string readStringLn(int l,int r){
5728     return readString(l,r,'\n');
5729 }
5730 string readStringSp(int l,int r){
5731     return readString(l,r,' ');
5732 }
5733 int sumN = 0;
5734 void solve()
5735 {
5736     int N = readInt(1, 100000, '\n');
5737     sumN += N;
5738     assert(sumN <= 5e5);
5739     int A[N+1];
5740     ll total = 0;
5741     int mini = 1e9;

```

```

5742     for(int i = 1; i <= N; i++)
5743     {
5744         if(i==N)
5745             A[i] = readInt(1, 1000000000, '\n');
5746         else
5747             A[i] = readInt(1, 1000000000, ' ');
5748         total += A[i];
5749         mini = min(mini, A[i]);
5750     }
5751     if(N%2 == 1)
5752     {
5753         if(total % 2 == 1)
5754             cout << "CHEF\n";
5755         else
5756             cout << "CHEFINA\n";
5757         return;
5758     }
5759     if(total%2 == 1 || mini%2 == 1)
5760     {
5761         cout << "CHEF\n";
5762         return;
5763     }
5764     if(mini%2 == 0)
5765     {
5766         cout << "CHEFINA\n";
5767         return;
5768     }
5769 }
5770 int main()
5771 {
5772     #ifndef ONLINE_JUDGE
5773     freopen("input.txt", "r", stdin);
5774     freopen("output.txt", "w", stdout);
5775     #endif
5776     ios_base::sync_with_stdio(false);
5777     cin.tie(NULL), cout.tie(NULL);
5778     int T=readInt(1,5000,'\n');
5779     while(T--)
5780         solve();
5781     assert(getchar()==-1);
5782     cerr << "Time : " << 1000 * ((double)clock()) / (double)CLOCKS_PER_SEC << "ms\n";
5783 }
5784
5785 //OR_XOR
5786 #include <bits/stdc++.h>
5787 using namespace std;
5788
5789 #include <ext/pb_ds/assoc_container.hpp>
5790 #include <ext/pb_ds/tree_policy.hpp>
5791
5792 typedef long long int ll;
5793
5794 using namespace __gnu_pbds;
5795 template <typename T> using ordered_set = tree<T, null_type, less<T>, rb_tree_tag,
tree_order_statistics_node_update>;
5796 // ordered_set -> find_by_order(x)<itr, x being 0-indexed>; order_of_key(x)<count,
strictly less>
5797
5798
5799 #define int ll
5800 #define fast ios::sync_with_stdio(0),cin.tie(0), cout.tie(0);
5801 #define rep(i, m, n) for (ll i = m; i < n; i++)
5802 #define ppi pair<int, int>
5803 #define pb push_back
5804 #define endl "\n"
5805 #define all(v) (v).begin(), (v).end()
5806 #define f first
5807 #define ss second
5808 #define in insert

```

```

5809 #define lb lower_bound
5810 #define ub upper_bound
5811 #define sz size()
5812 #define bg begin()
5813 #define pq priority_queue
5814 #define vc vector<int>
5815 #define vcp vector<ppi>
5816 #define mp map<int, int>
5817 #define gp gp_hash_table<int, int, chash>
5818 #define meml(a) memset(a, -1, sizeof(a));
5819 #define memt(a) memset(a, true, sizeof(a));
5820 #define re(a) {cout<<a<<endl;}
5821 // #define re(a) return a;
5822 #define sd greater<int>()
5823 #define sdp greater<ppi>()
5824 #define enl "\n"; return;
5825 // #define SET(n) cout << fixed << setprecision(n)
5826 #define ppc __builtin_popcountll
5827 #ifndef ONLINE_JUDGE
5828 #define debug(x) cerr << #x <<" : "; _print(x); cerr << endl;
5829 #else
5830 #define debug(x)
5831 #endif
5832
5833 template<typename T> istream& operator>>(istream& is, vector<T> &v){ for(auto& i : v)
is >> i; return is;}
5834 template<typename T> ostream& operator<<(ostream& os, vector<T> v){for (auto& i : v)
os << i << ' '; return os;}
5835
5836 template<class T> void _print(T n){cerr<<n;}
5837 template<class T, class V> void _print(T a[], V n){cerr<<"Array: [ "; rep(i, 0, n){
_print(a[i]); cerr<<" ";} cerr<<" ] \n";}
5838 template<class T, class V> void _print(pair<T, T> a[], V n){cerr<<"Pair Array: [ "; rep(
i, 0, n){cerr<<"{";_print(a[i].f); cerr<<" , "; _print(a[i].ss); cerr<<"},";cerr<<" ";}
cerr<<"] \n";}
5839 template <class T, class V> void _print(pair <T, V> p) {cerr << "{"; _print(p.f); cerr
<< " , "; _print(p.ss); cerr << " }";}
5840 template <class T> void _print(vector <T> v) {cerr << "[ "; for (T i : v) {_print(i);
cerr << " ";} cerr << " ]";}
5841 template <class T> void _print(set <T> v) {cerr << "[ "; for (T i : v) {_print(i); cerr
<< " ";} cerr << " ]";}
5842 template <class T, class V> void _print(map <T, V> v) {cerr << "[ "; for (auto i : v) {
_print(i); cerr << " ";} cerr << " ]";}
5843 const double eps=1e-6;
5844 const int MOD=1e9+7, inf=INT_MAX, inff=INT_MIN;
5845 //998244353
5846 const int N=(1e5)+5;
5847 const int RANDOM = chrono::high_resolution_clock::now().time_since_epoch().count();
5848 struct chash { // To use most bits rather than just the lowest ones:
5849     int MUL=1e9+3;
5850     int operator()(int x) const { return std::hash<ll>{}((x ^ RANDOM) % MOD * MUL); }
5851 };
5852 ll expol(ll a, ll b) {ll res = 1; while (b > 0) { if (b & 1)res = (res * a); a =
(a * a); b = b >> 1;} return res;}
5853 ll expo(ll a, ll b, ll MOD=1e9+7) {ll res = 1; a%=MOD; while (b > 0) {if (b & 1)res =
(res * a) % MOD; a = (a * a) % MOD; b = b >> 1;} return res;}
5854 int LOG(ll n, ll x) {int ans=-1;while(n>0){ ans++, n/=x;}return ans;}
5855 int Ceil(ll a, ll b) {if(a%b==0 || a<0) return a/b; else return a/b+1;}
5856 int dx[]={1, 0, -1, 0}, dy[]={0, -1, 0, 1};
5857
5858 int Solve(vector<int>&a)
5859 {
5860     int n=a.size();
5861     vector<int> prefix(n);
5862
5863     map<int, vector<int>>m;
5864     vector<int>last(31, -1);
5865     int xo=0, ans=0;
5866     m[0].pb(-1);

```

```

5867
5868     rep(i, 0, n)
5869     {
5870         for(int j=0; j<31; j++)
5871         {
5872             if((a[i]>>j)&1) last[j]=i;
5873         }
5874         xo^=a[i];
5875         prefix[i]=xo;
5876
5877         vector<int>t=last;
5878         sort(all(t), greater<int>());
5879
5880         int OR=a[i], past=i;
5881
5882         for(int j=0; j<31; j++)
5883         {
5884             if((j && t[j]==t[j-1]) || t[j]==i) continue;
5885             int k=t[j];
5886             int x=(xo^OR);
5887             auto it=lb(all(m[x]), min(past, i-1))-lb(all(m[x]), k);
5888             ans+=it;
5889             OR|=a[k];
5890             past=k;
5891         }
5892         m[xo].pb(i);
5893     }
5894     return n*(n-1)/2-ans;
5895 }
5896
5897 signed main()
5898 {
5899     fast
5900     #ifndef ONLINE_JUDGE
5901     freopen("Error.txt", "w", stderr);
5902     #endif
5903
5904
5905     int T;
5906     cin >> T;
5907     int i=1;
5908
5909     while(T--)
5910     {
5911         int n; cin>>n;
5912         vc v(n); cin>>v;
5913         cout<<Solve(v)<<endl;
5914     }
5915
5916     #ifndef ONLINE_JUDGE
5917     cerr<<"\nTime taken : "<<(float)clock()/CLOCKS_PER_SEC<<" secs"<<"\n";
5918     #endif
5919
5920     return 0;
5921 }
5922
5923 //CHECKPOINT
5924 //Utkarsh.25dec
5925 #include <iostream>
5926 #include <cstdio>
5927 #include <cstdlib>
5928 #include <algorithm>
5929 #include <cmath>
5930 #include <vector>
5931 #include <set>
5932 #include <map>
5933 #include <unordered_set>
5934 #include <unordered_map>
5935 #include <queue>

```

```

5936 #include <ctime>
5937 #include <cassert>
5938 #include <complex>
5939 #include <string>
5940 #include <cstring>
5941 #include <chrono>
5942 #include <random>
5943 #include <bitset>
5944 #include <array>
5945 #define ll long long int
5946 #define pb push_back
5947 #define mp make_pair
5948 #define mod 998244353
5949 #define vl vector<ll>
5950 #define all(c) (c).begin(), (c).end()
5951 using namespace std;
5952 typedef vector<vector<ll>> matrix;
5953 ll power(ll a,ll b) {ll res=1;a%=mod; assert(b>=0); for(;b;b>>=1){if(b&1) res=res*a%mod;a
=a*a%mod;}return res;}
5954 ll modInverse(ll a){return power(a,mod-2);}
5955 const int N=500023;
5956 bool vis[N];
5957 vector<int> adj[N];
5958
5959 long long readInt(long long l,long long r,char endd){
5960     long long x=0;
5961     int cnt=0;
5962     int fi=-1;
5963     bool is_neg=false;
5964     while(true){
5965         char g=getchar();
5966         if(g=='-'){
5967             assert(fi===-1);
5968             is_neg=true;
5969             continue;
5970         }
5971         if('0'<=g && g<='9'){
5972             x*=10;
5973             x+=g-'0';
5974             if(cnt==0){
5975                 fi=g-'0';
5976             }
5977             cnt++;
5978             assert(fi!=0 || cnt==1);
5979             assert(fi!=0 || is_neg==false);
5980
5981             assert(!(cnt>19 || ( cnt==19 && fi>1) ));
5982         } else if(g==endd){
5983             if(is_neg){
5984                 x= -x;
5985             }
5986
5987             if(!(l <= x && x <= r))
5988             {
5989                 cerr << l << ' ' << r << ' ' << x << '\n';
5990                 assert(1 == 0);
5991             }
5992
5993             return x;
5994         } else {
5995             assert(false);
5996         }
5997     }
5998 }
5999 string readString(int l,int r,char endd){
6000     string ret="";
6001     int cnt=0;
6002     while(true){
6003         char g=getchar();

```



```

6004         assert(g!=-1);
6005         if(g==endd){
6006             break;
6007         }
6008         cnt++;
6009         ret+=g;
6010     }
6011     assert(l<=cnt && cnt<=r);
6012     return ret;
6013 }
6014 long long readIntSp(long long l,long long r){
6015     return readInt(l,r,' ');
6016 }
6017 long long readIntLn(long long l,long long r){
6018     return readInt(l,r,'\n');
6019 }
6020 string readStringLn(int l,int r){
6021     return readString(l,r,'\n');
6022 }
6023 string readStringSp(int l,int r){
6024     return readString(l,r,' ');
6025 }
6026
6027 const int K = 2;
6028 // computes A * B
6029 matrix mul(matrix A, matrix B)
6030 {
6031     matrix C(K+1, vector<ll>(K+1));
6032     for(int i=1;i<=K;i++) for(int j=1;j<=K;j++) for(int k=1;k<=K;k++)
6033         C[i][j] = (C[i][j] + A[i][k] * B[k][j]) % mod;
6034     return C;
6035 }
6036
6037 // computes A ^ p
6038 matrix pow(matrix A, ll p)
6039 {
6040     if (p == 1)
6041         return A;
6042     if (p % 2)
6043         return mul(A, pow(A, p-1));
6044     matrix X = pow(A, p/2);
6045     return mul(X, X);
6046 }
6047 //matrix ans(K+1,vl(K+1));
6048
6049 matrix ans(K+1, vl(K+1));
6050 void solve()
6051 {
6052     ll N, M;
6053     N = readInt(1, 1000000000, ' ');
6054     M = readInt(2, 100000, '\n');
6055     ll x;
6056     if(M%2==0)
6057         x = modInverse(2);
6058     else
6059         x = ((M+1)/2 * modInverse(M))%mod;
6060     ans[1][1] = (2*x-1+mod)%mod;
6061     ans[1][2] = (1 - x + mod)%mod;
6062     ans[2][1] = 0;
6063     ans[2][2] = 1;
6064     if (N==1)
6065     {
6066         cout<<x<<'\n';
6067         return;
6068     }
6069     ans = pow(ans, N-1);
6070     ll out = (ans[1][1]*x + ans[1][2])%mod;
6071     cout << out << '\n';
6072 }

```

```

6073 int main()
6074 {
6075     #ifndef ONLINE_JUDGE
6076     freopen("input.txt", "r", stdin);
6077     freopen("output.txt", "w", stdout);
6078     #endif
6079     ios_base::sync_with_stdio(false);
6080     cin.tie(NULL), cout.tie(NULL);
6081     int T=readInt(1,100000,'\n');
6082     while(T--)
6083         solve();
6084     cerr << "Time : " << 1000 * ((double)clock()) / (double)CLOCKS_PER_SEC << "ms\n";
6085 }
6086
6087 //SUBSEQINV
6088 //Utkarsh.25dec
6089 #include <iostream>
6090 #include <cstdio>
6091 #include <cstdlib>
6092 #include <algorithm>
6093 #include <cmath>
6094 #include <vector>
6095 #include <set>
6096 #include <map>
6097 #include <unordered_set>
6098 #include <unordered_map>
6099 #include <queue>
6100 #include <ctime>
6101 #include <cassert>
6102 #include <complex>
6103 #include <string>
6104 #include <cstring>
6105 #include <chrono>
6106 #include <random>
6107 #include <bitset>
6108 #include <array>
6109 #define ll long long int
6110 #define pb push_back
6111 #define mp make_pair
6112 #define mod 998244353
6113 #define vl vector<ll>
6114 #define all(c) (c).begin(), (c).end()
6115 using namespace std;
6116 ll power(ll a,ll b) {ll res=1;a%=mod; assert(b>=0); for(;b;b>=1){if(b&1)res=res*a%mod;a
=a*a%mod;}return res;}
6117 ll modInverse(ll a){return power(a,mod-2);}
6118 const int N=500023;
6119 bool vis[N];
6120 vector<int> adj[N];
6121 long long readInt(long long l,long long r,char endd){
6122     long long x=0;
6123     int cnt=0;
6124     int fi=-1;
6125     bool is_neg=false;
6126     while(true){
6127         char g=getchar();
6128         if(g=='-'){
6129             assert(fi===-1);
6130             is_neg=true;
6131             continue;
6132         }
6133         if('0'<=g && g<='9'){
6134             x*=10;
6135             x+=g-'0';
6136             if(cnt==0){
6137                 fi=g-'0';
6138             }
6139             cnt++;
6140             assert(fi!=0 || cnt==1);

```

```

6141         assert(fi!=0 || is_neg==false);
6142
6143         assert(!(cnt>19 || ( cnt==19 && fi>1) ));
6144     } else if(g==endd){
6145         if(is_neg){
6146             x= -x;
6147         }
6148
6149         if(!(l <= x && x <= r))
6150         {
6151             cerr << l << ' ' << r << ' ' << x << '\n';
6152             assert(1 == 0);
6153         }
6154
6155         return x;
6156     } else {
6157         assert(false);
6158     }
6159 }
6160
6161 string readString(int l,int r,char endd){
6162     string ret="";
6163     int cnt=0;
6164     while(true){
6165         char g=getchar();
6166         assert(g!=-1);
6167         if(g==endd){
6168             break;
6169         }
6170         cnt++;
6171         ret+=g;
6172     }
6173     assert(l<=cnt && cnt<=r);
6174     return ret;
6175 }
6176 long long readIntSp(long long l,long long r){
6177     return readInt(l,r,' ');
6178 }
6179 long long readIntLn(long long l,long long r){
6180     return readInt(l,r,'\n');
6181 }
6182 string readStringLn(int l,int r){
6183     return readString(l,r,'\n');
6184 }
6185 string readStringSp(int l,int r){
6186     return readString(l,r,' ');
6187 }
6188 int sumN = 0;
6189 void solve()
6190 {
6191     int N = readInt(1,100000,'\n');
6192     sumN+=N;
6193     assert(sumN<=500000);
6194     int A[N+1];
6195     set <int> s;
6196     for(int i=1;i<=N;i++)
6197     {
6198         if(i==N)
6199             A[i] = readInt(1, N, '\n');
6200         else
6201             A[i] = readInt(1, N, ' ');
6202         s.insert(A[i]);
6203     }
6204     assert(s.size() == N);
6205     int prefMax[N+10];
6206     int suffMin[N+10];
6207     prefMax[0] = 0;
6208     suffMin[N+1] = N+1;
6209     for(int i=1;i<=N;i++)

```

```

6210     prefMax[i] = max(prefMax[i-1],A[i]);
6211     for(int i=N;i>=1;i--)
6212         suffMin[i] = min(suffMin[i+1],A[i]);
6213     int necessary = 0;
6214     for(int i=1;i<=N;i++)
6215     {
6216         if(A[i]<prefMax[i-1] || A[i]>suffMin[i+1])
6217             necessary++;
6218     }
6219     ll ans = power(2, N-necessary);
6220     if(necessary == 0)
6221         ans = (ans + mod - 1)%mod;
6222     cout<<ans<<"\n";
6223 }
6224 int main()
6225 {
6226     #ifndef ONLINE_JUDGE
6227     freopen("input.txt", "r", stdin);
6228     freopen("output.txt", "w", stdout);
6229     #endif
6230     ios_base::sync_with_stdio(false);
6231     cin.tie(NULL),cout.tie(NULL);
6232     int T=readInt(1,1000,'\n');
6233     while(T--)
6234         solve();
6235     cerr << "Time : " << 1000 * ((double)clock()) / (double)CLOCKS_PER_SEC << "ms\n";
6236 }
6237
6238 //LEASTSIZE
6239
6240 #include <bits/stdc++.h>
6241 #include <ext/pb_ds/tree_policy.hpp>
6242 #include <ext/pb_ds/assoc_container.hpp>
6243 using namespace __gnu_pbds;
6244 using namespace std;
6245 #define ll long long
6246 const ll INF_MUL=1e13;
6247 const ll INF_ADD=1e18;
6248 #define pb push_back
6249 #define mp make_pair
6250 #define nline "\n"
6251 #define f first
6252 #define s second
6253 #define pll pair<ll,ll>
6254 #define all(x) x.begin(),x.end()
6255 #define vl vector<ll>
6256 #define vvl vector<vector<ll>>
6257 #define vvv vector<vector<vector<ll>>>
6258 #ifndef ONLINE_JUDGE
6259 #define debug(x) cerr<<#x<<" "; _print(x); cerr<<nline;
6260 #else
6261 #define debug(x);
6262 #endif
6263 void _print(ll x){cerr<<x;}
6264 void _print(int x){cerr<<x;}
6265 void _print(char x){cerr<<x;}
6266 void _print(string x){cerr<<x;}
6267 mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());
6268 template<class T,class V> void _print(pair<T,V> p) {cerr<<"{"; _print(p.first);cerr<<",";
6269 _print(p.second);cerr<<"}";}
6269 template<class T>void _print(vector<T> v) {cerr<<" [ "; for (T i:v){_print(i);cerr<<" ";
6270 ;}cerr<<"]";}
6270 template<class T>void _print(set<T> v) {cerr<<" [ "; for (T i:v){_print(i); cerr<<" ";}
6271 cerr<<"]";}
6271 template<class T>void _print(multiset<T> v) {cerr<<" [ "; for (T i:v){_print(i);cerr<<
6272 " ";}cerr<<"]";}
6272 template<class T,class V>void _print(map<T, V> v) {cerr<<" [ "; for(auto i:v) {_print(i
6273 );cerr<<" ";} cerr<<"]";}
6273 typedef tree<ll, null_type, less<ll>, rb_tree_tag, tree_order_statistics_node_update>

```

```

ordered_set;
6274 typedef tree<ll, null_type, less_equal<ll>, rb_tree_tag,
tree_order_statistics_node_update> ordered_multiset;
6275 typedef tree<pair<ll, ll>, null_type, less<pair<ll, ll>>, rb_tree_tag,
tree_order_statistics_node_update> ordered_pset;
6276 //-----
-----
6277 const ll MOD=1e9+7;
6278 const ll MAX=100100;
6279 ll root, n;
6280 vector<ll> adj[MAX];
6281 vector<ll> subtree(MAX);
6282 ll now=0, till=log2(MAX)+1;
6283 vector<ll> tin(MAX, 0), tout(MAX, 0), depth(MAX, 0);
6284 vector<vector<ll>> jump(MAX, vector<ll>(till+1, 0));
6285 void dfs(ll cur, ll par) {
6286     jump[cur][0]=par;
6287     for(ll i=1; i<=till; i++)
6288         jump[cur][i]=jump[jump[cur][i-1]][i-1];
6289     tin[cur]=++now;
6290     for(ll chld:adj[cur]){
6291         if(chld!=par){
6292             depth[chld]=depth[cur]+1;
6293             dfs(chld, cur);
6294         }
6295     }
6296     tout[cur]=++now;
6297 }
6298 bool is_ancestor(ll a, ll b){
6299     if((tin[a]<=tin[b])&&(tout[a]>=tout[b]))
6300         return 1;
6301     return 0;
6302 }
6303 ll lca(ll a, ll b){
6304     if(is_ancestor(a, b))
6305         return a;
6306     for(ll i=till; i>=0; i--){
6307         if(!is_ancestor(jump[a][i], b))
6308             a=jump[a][i];
6309     }
6310     return jump[a][0];
6311 }
6312 void dfs_1(ll cur, ll par){
6313     subtree[cur]=1;
6314     ll nax=0;
6315     for(auto chld:adj[cur]){
6316         if(chld!=par){
6317             dfs_1(chld, cur);
6318             nax=max(nax, subtree[chld]);
6319             subtree[cur]+=subtree[chld];
6320         }
6321     }
6322     nax=max(nax, n-subtree[cur]);
6323     if(2*nax <= n){
6324         root=cur;
6325     }
6326 }
6327 vector<ll> track(MAX, 0);
6328 void dfs_2(ll cur, ll par){
6329     for(auto chld:adj[cur]){
6330         if(chld!=par){
6331             if(cur==root){
6332                 track[chld]=chld;
6333             }
6334             else{
6335                 track[chld]=track[cur];
6336             }
6337             dfs_2(chld, cur);

```

```

6338     }
6339 }
6340 }
6341 void solve() {
6342     cin >> n;
6343     for (ll i = 1; i <= n; i++) {
6344         ll u, v; cin >> u >> v;
6345         adj[u].push_back(v);
6346         adj[v].push_back(u);
6347     }
6348     dfs_1(1, -1);
6349     track[root] = root;
6350     dfs_2(root, -1);
6351     dfs(root, root);
6352     vector<pair<ll, ll>> order;
6353     for (ll i = 1; i <= n; i++) {
6354         order.push_back({track[i], i});
6355     }
6356     sort(all(order));
6357     cout << root << endl;
6358     vector<set<ll>> use(n + 5);
6359     for (ll i = 1; i <= n; i++) {
6360         use[track[i]].insert(i);
6361     }
6362     set<pair<ll, ll>> comp;
6363     for (ll i = 1; i <= n; i++) {
6364         comp.insert({use[i].size(), i});
6365         adj[i].clear();
6366     }
6367     debug(comp);
6368     ll prv = -1;
6369     for (ll i = 1; i <= n; i++) {
6370         auto it = --comp.end();
6371         ll node;
6372         while (1) {
6373             if ((*it).s != prv) {
6374                 break;
6375             }
6376             it--;
6377         }
6378         node = (*it).s;
6379         debug(mp(i, node));
6380         comp.erase({use[node].size(), node});
6381         cout << *use[node].begin() << " \n" [i == n];
6382         use[node].erase(use[node].begin());
6383         comp.insert({use[node].size(), node});
6384         prv = node;
6385     }
6386     return;
6387 }
6388 int main()
6389 {
6390     ios_base::sync_with_stdio(false);
6391     cin.tie(NULL);
6392     #ifndef ONLINE_JUDGE
6393     freopen("input.txt", "r", stdin);
6394     freopen("output.txt", "w", stdout);
6395     freopen("error.txt", "w", stderr);
6396     #endif
6397     ll test_cases = 1;
6398     cin >> test_cases;
6399     while (test_cases--) {
6400         solve();
6401     }
6402     cout << fixed << setprecision(10);
6403     cerr << "Time: " << 1000 * ((double)clock()) / ((double)CLOCKS_PER_SEC) << "ms\n";
6404 }

```

```

6405
6406 //MINIMUMOP
6407 #pragma GCC optimization("O3")
6408 #pragma GCC optimization("Ofast,unroll-loops")
6409
6410 #include <bits/stdc++.h>
6411 #include <ext/pb_ds/tree_policy.hpp>
6412 #include <ext/pb_ds/assoc_container.hpp>
6413 using namespace __gnu_pbds;
6414 using namespace std;
6415 #define ll long long
6416 const ll INF_MUL=1e13;
6417 const ll INF_ADD=1e18;
6418 #define pb push_back
6419 #define mp make_pair
6420 #define nline "\n"
6421 #define f first
6422 #define s second
6423 #define pll pair<ll,ll>
6424 #define all(x) x.begin(),x.end()
6425 #define vl vector<ll>
6426 #define vvl vector<vector<ll>>
6427 #define vvvl vector<vector<vector<ll>>>
6428 #ifndef ONLINE_JUDGE
6429 #define debug(x) cerr<<#x<<" "; _print(x); cerr<<nline;
6430 #else
6431 #define debug(x);
6432 #endif
6433 void _print(ll x){cerr<<x;}
6434 void _print(int x){cerr<<x;}
6435 void _print(char x){cerr<<x;}
6436 void _print(string x){cerr<<x;}
6437 mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());
6438 template<class T,class V> void _print(pair<T,V> p) {cerr<<"{"; _print(p.first);cerr<<" ";
6439   _print(p.second);cerr<<"}";}
6439 template<class T>void _print(vector<T> v) {cerr<<" [ "; for (T i:v){_print(i);cerr<<" "
6440   ;}cerr<<" ]";}
6440 template<class T>void _print(set<T> v) {cerr<<" [ "; for (T i:v){_print(i); cerr<<" ";}
6441   cerr<<" ]";}
6441 template<class T>void _print(multiset<T> v) {cerr<<" [ "; for (T i:v){_print(i);cerr<<"
6442   " ";}cerr<<" ]";}
6442 template<class T,class V>void _print(map<T, V> v) {cerr<<" [ "; for(auto i:v) {_print(i
6443   );cerr<<" ";} cerr<<" ]";}
6443 typedef tree<ll, null_type, less<ll>, rb_tree_tag, tree_order_statistics_node_update>
6444   ordered_set;
6444 typedef tree<ll, null_type, less_equal<ll>, rb_tree_tag,
6445   tree_order_statistics_node_update> ordered_multiset;
6445 typedef tree<pair<ll,ll>, null_type, less<pair<ll,ll>>, rb_tree_tag,
6446   tree_order_statistics_node_update> ordered_pset;
6446 //-----
6447 -----
6447 const ll MOD=1e9+7;
6448 const ll MAX=1000500;
6449 vector<ll> freq(MAX,0);
6450 vector<ll> prime(MAX,1);
6451 vector<vector<ll>> track(MAX);
6452 void solve(){
6453     ll n,m; cin>>n>>m;
6454     vector<ll> a(n);
6455     for(auto &it:a){
6456         cin>>it;
6457     }
6458     sort(all(a));
6459     if(a[0]==a[n-1]){
6460         cout<<0<<nline;
6461         return;
6462     }
6463     for(auto it:a){

```

```

6464         for(auto i:track[it]){
6465             freq[i]++;
6466         }
6467     }
6468     auto init=[&]() {
6469         for(auto it:a){
6470             for(auto i:track[it]){
6471                 freq[i]=0;
6472             }
6473         }
6474     };
6475     for(ll i=2;i<=m;i++){
6476         if(!prime[i]){
6477             continue;
6478         }
6479         if(freq[i]==0 or freq[i]==n){
6480             init();
6481             cout<<1<<endl<<i<<endl;
6482             return;
6483         }
6484     }
6485     init();
6486     cout<<2<<endl<<2<<" "<<3<<endl;
6487     return;
6488 }
6489 int main()

6490 {
6491     ios_base::sync_with_stdio(false);
6492     cin.tie(NULL);
6493     #ifndef ONLINE_JUDGE
6494     freopen("input.txt", "r", stdin);
6495     freopen("output.txt", "w", stdout);
6496     freopen("error.txt", "w", stderr);
6497     #endif
6498     ll test_cases=1;
6499     cin>>test_cases;
6500     prime[1]=0;
6501     for(ll i=2;i<MAX;i++){
6502         if(!prime[i]){
6503             continue;
6504         }
6505         for(ll j=i+i;j<MAX;j+=i){
6506             prime[j]=0;
6507             track[j].push_back(i);
6508         }
6509         track[i].push_back(i);
6510     }
6511     while(test_cases--){
6512         solve();
6513     }
6514     cout<<fixed<<setprecision(10);
6515     cerr<<"Time:"<<1000*((double)clock())/(double)CLOCKS_PER_SEC<<"ms\n";
6516 }

6517
6518 //LARGESTY
6519 #pragma GCC optimization("O3")
6520 #pragma GCC optimization("Ofast,unroll-loops")
6521
6522 #include <bits/stdc++.h>
6523 #include <ext/pb_ds/tree_policy.hpp>
6524 #include <ext/pb_ds/assoc_container.hpp>
6525 using namespace __gnu_pbds;
6526 using namespace std;
6527 #define ll long long
6528 const ll INF_ADD=1e18;
6529 #define pb push_back
6530 #define mp make_pair

```



```

6531 #define nline "\n"
6532 #define f first
6533 #define s second
6534 #define pll pair<ll,ll>
6535 #define all(x) x.begin(),x.end()
6536 #define vl vector<ll>
6537 #define vvl vector<vector<ll>>
6538 #define vvv vector<vector<vector<ll>>>
6539 #ifndef ONLINE_JUDGE
6540 #define debug(x) cerr<<"x<<" "; _print(x); cerr<<nline;
6541 #else
6542 #define debug(x);
6543 #endif
6544 void _print(ll x){cerr<<x;}
6545 void _print(int x){cerr<<x;}
6546 void _print(char x){cerr<<x;}
6547 void _print(string x){cerr<<x;}
6548 mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());
6549 template<class T,class V> void _print(pair<T,V> p) {cerr<<"{"; _print(p.first);cerr<<" ";
; _print(p.second);cerr<<""}";}
6550 template<class T>void _print(vector<T> v) {cerr<<" [ "; for (T i:v){_print(i);cerr<<" "
;};cerr<<" ]";}
6551 template<class T>void _print(set<T> v) {cerr<<" [ "; for (T i:v){_print(i); cerr<<" ";}
cerr<<" ]";}
6552 template<class T>void _print(multiset<T> v) {cerr<<" [ "; for (T i:v){_print(i);cerr<<"
";};cerr<<" ]";}
6553 template<class T,class V>void _print(map<T, V> v) {cerr<<" [ "; for(auto i:v) {_print(i
);cerr<<" ";} cerr<<" ]";}
6554 typedef tree<ll, null_type, less<ll>, rb_tree_tag, tree_order_statistics_node_update>
ordered_set;
6555 typedef tree<ll, null_type, less_equal<ll>, rb_tree_tag,
tree_order_statistics_node_update> ordered_multiset;
6556 typedef tree<pair<ll,ll>, null_type, less<pair<ll,ll>>, rb_tree_tag,
tree_order_statistics_node_update> ordered_pset;
6557 //-----
-----
6558 const ll MOD=1e9+7;
6559 const ll MAX=200200;
6560 void solve(){
6561     ll n,x; cin>>n>>x;
6562     vector<ll> a(n);
6563     for(auto &it:a){
6564         cin>>it;
6565     }
6566     ll smallest=0;
6567     set<ll> need;
6568     for(ll i=29;i>=0;i--){
6569         ll found=0;
6570         for(auto it:a){
6571             found+=min(1ll,it&(1<<i));
6572         }
6573         if(found>=1 and found!=n){
6574             need.insert(i);
6575         }
6576     }
6577     assert(need.size()>=1);
6578     smallest=*need.begin();
6579     ll ans=0,done=0;
6580     for(ll i=29;i>=0;i--){
6581         if(done or smallest<i){
6582             if(ans+(1<<i) <= x){
6583                 ans+=(1<<i);
6584             }
6585             else if(need.count(i)){
6586                 done=1;
6587             }
6588         }
6589     }

```

```

6590     vector<ll> b=a;
6591     for(auto &it:b){
6592         it|=ans;
6593     }
6594     sort(all(b));
6595     assert(b[0]!=b[n-1]);
6596     cout<<ans<<endl;
6597     return;
6598 }
6599 int main()

6600 {
6601     ios_base::sync_with_stdio(false);
6602     cin.tie(NULL);
6603     #ifndef ONLINE_JUDGE
6604     freopen("input.txt", "r", stdin);
6605     freopen("output.txt", "w", stdout);
6606     freopen("error.txt", "w", stderr);
6607     #endif
6608     ll test_cases=1;
6609     cin>>test_cases;
6610     while(test_cases--){
6611         solve();
6612     }
6613     cout<<fixed<<setprecision(10);
6614     cerr<<"Time:"<<1000*((double)clock())/((double)CLOCKS_PER_SEC<<"ms\n";
6615 }
6616
6617 //PREFIXMAX
6618 #pragma GCC optimization("O3")
6619 #pragma GCC optimization("Ofast,unroll-loops")
6620
6621 #include <bits/stdc++.h>
6622 #include <ext/pb_ds/tree_policy.hpp>
6623 #include <ext/pb_ds/assoc_container.hpp>
6624 using namespace __gnu_pbds;
6625 using namespace std;
6626 #define ll long long
6627 const ll INF_MUL=1e13;
6628 const ll INF_ADD=1e18;
6629 #define pb push_back
6630 #define mp make_pair
6631 #define nline "\n"
6632 #define f first
6633 #define s second
6634 #define pll pair<ll,ll>
6635 #define all(x) x.begin(),x.end()
6636 #define vl vector<ll>
6637 #define vvl vector<vector<ll>>
6638 #define vvvl vector<vector<vector<ll>>>
6639 #ifndef ONLINE_JUDGE
6640 #define debug(x) cerr<<#x<<" "; _print(x); cerr<<endl;
6641 #else
6642 #define debug(x);
6643 #endif
6644 void _print(ll x){cerr<<x;}
6645 void _print(int x){cerr<<x;}
6646 void _print(char x){cerr<<x;}
6647 void _print(string x){cerr<<x;}
6648 mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());
6649 template<class T,class V> void _print(pair<T,V> p){cerr<<"{"; _print(p.first);cerr<<" ";
6650 _print(p.second);cerr<<"}";}
6651 template<class T>void _print(vector<T> v){cerr<<" [ "; for (T i:v){_print(i);cerr<<" ";
6652 ;}cerr<<" ]";}
6651 template<class T>void _print(set<T> v){cerr<<" [ "; for (T i:v){_print(i); cerr<<" ";
6652 cerr<<" ]";}
6652 template<class T>void _print(multiset<T> v){cerr<<" [ "; for (T i:v){_print(i);cerr<<
6652 " ";}cerr<<" ]";}

```

```

6653     template<class T, class V> void _print(map<T, V> v) {cerr<<" [ "; for(auto i:v) {_print(i
6654 );cerr<<" ";} cerr<<"]";}
6655     typedef tree<ll, null_type, less<ll>, rb_tree_tag, tree_order_statistics_node_update>
ordered_set;
6656     typedef tree<ll, null_type, less_equal<ll>, rb_tree_tag,
tree_order_statistics_node_update> ordered_multiset;
6657     typedef tree<pair<ll, ll>, null_type, less<pair<ll, ll>>, rb_tree_tag,
tree_order_statistics_node_update> ordered_pset;
6658     //-----
6659     -----
6660     const ll MOD=998244353;
6661     const ll MAX=2000200;
6662     vector<ll> fact(MAX+2, 1), inv_fact(MAX+2, 1);
6663     ll binpow(ll a, ll b, ll MOD) {
6664         ll ans=1;
6665         a%=MOD;
6666         while(b) {
6667             if(b&1)
6668                 ans=(ans*a)%MOD;
6669             b/=2;
6670             a=(a*a)%MOD;
6671         }
6672         return ans;
6673     }
6674     ll inverse(ll a, ll MOD) {
6675         return binpow(a, MOD-2, MOD);
6676     }
6677     void precompute(ll MOD) {
6678         for(ll i=2; i<MAX; i++) {
6679             fact[i]=(fact[i-1]*i)%MOD;
6680         }
6681         inv_fact[MAX-1]=inverse(fact[MAX-1], MOD);
6682         for(ll i=MAX-2; i>=0; i--) {
6683             inv_fact[i]=(inv_fact[i+1]*(i+1))%MOD;
6684         }
6685     }
6686     ll nCr(ll a, ll b, ll MOD) {
6687         if((a<0) || (a<b) || (b<0))
6688             return 0;
6689         ll denom=(inv_fact[b]*inv_fact[a-b])%MOD;
6690         return (fact[a]*denom)%MOD;
6691     }
6692     void solve() {
6693         ll n; cin>>n;
6694         ll nax=0;
6695         for(ll i=1; i<=n; i++) {
6696             ll x; cin>>x;
6697             nax=max(nax, x);
6698         }
6699         ll x=n, y=nax;
6700         ll ans=nCr(4*n-x-y, 2*n-y, MOD);
6701         ll a=y+1, b=x-1;
6702         x=a, y=b;
6703         ans=(ans-nCr(4*n-x-y, 2*n-y, MOD)+MOD)%MOD;
6704         cout<<ans<<endl;
6705         return;
6706     }
6707     int main()
6708
6709 {
6710     ios_base::sync_with_stdio(false);
6711     cin.tie(NULL);
6712     #ifndef ONLINE_JUDGE
6713     freopen("input.txt", "r", stdin);
6714     freopen("output.txt", "w", stdout);
6715     freopen("error.txt", "w", stderr);
6716     #endif

```

```

6714     ll test_cases=1;
6715     cin>>test_cases;
6716     precompute(MOD);
6717     while(test_cases--){
6718         solve();
6719     }
6720     cout<<fixed<<setprecision(10);
6721     cerr<<"Time:"<<1000*((double)clock())/(double)CLOCKS_PER_SEC<<"ms\n";
6722 }
6723
6724 //SUMOVERALL
6725 #pragma GCC optimization("O3")
6726 #pragma GCC optimization("Ofast,unroll-loops")
6727
6728 #include <bits/stdc++.h>
6729 #include <ext/pb_ds/tree_policy.hpp>
6730 #include <ext/pb_ds/assoc_container.hpp>
6731 using namespace __gnu_pbds;
6732 using namespace std;
6733 #define ll long long
6734 const ll INF_MUL=1e13;
6735 const ll INF_ADD=1e18;
6736 #define pb push_back
6737 #define mp make_pair
6738 #define nline "\n"
6739 #define f first
6740 #define s second
6741 #define pll pair<ll,ll>
6742 #define all(x) x.begin(),x.end()
6743 #define vl vector<ll>
6744 #define vvl vector<vector<ll>>
6745 #define vvvl vector<vector<vector<ll>>>
6746 #ifndef ONLINE_JUDGE
6747 #define debug(x) cerr<<#x<<" "; _print(x); cerr<<nline;
6748 #else
6749 #define debug(x);
6750 #endif
6751 void _print(ll x){cerr<<x;}
6752 void _print(int x){cerr<<x;}
6753 void _print(char x){cerr<<x;}
6754 void _print(string x){cerr<<x;}
6755 mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());
6756 template<class T,class V> void _print(pair<T,V> p) {cerr<<"{"; _print(p.first);cerr<<" ";
6757 ; _print(p.second);cerr<<"}";}
6758 template<class T>void _print(vector<T> v) {cerr<<" [ "; for (T i:v){_print(i);cerr<<" "
6759 ;}cerr<<"]";}
6760 template<class T>void _print(set<T> v) {cerr<<" [ "; for (T i:v){_print(i); cerr<<" ";}
6761 cerr<<"]";}
6762 template<class T>void _print(multiset<T> v) {cerr<<" [ "; for (T i:v){_print(i);cerr<<
6763 " ";}cerr<<"]";}
6764 template<class T,class V>void _print(map<T, V> v) {cerr<<" [ "; for(auto i:v) {_print(i
6765 );cerr<<" ";} cerr<<"]";}
6766 typedef tree<ll, null_type, less<ll>, rb_tree_tag, tree_order_statistics_node_update>
6767 ordered_set;
6768 typedef tree<ll, null_type, less_equal<ll>, rb_tree_tag,
6769 tree_order_statistics_node_update> ordered_multiset;
6770 typedef tree<pair<ll,ll>, null_type, less<pair<ll,ll>>, rb_tree_tag,
6771 tree_order_statistics_node_update> ordered_pset;
6772 //-----
6773 -----
6774
6775 const ll MOD=998244353;
6776 const ll MAX=5005;
6777 vector<ll> fact(MAX+2,1),inv_fact(MAX+2,1);
6778 ll binpow(ll a,ll b,ll MOD){
6779     ll ans=1;
6780     a%=MOD;
6781     while(b){
6782         if(b&1)

```

```

6773         ans=(ans*a)%MOD;
6774         b/=2;
6775         a=(a*a)%MOD;
6776     }
6777     return ans;
6778 }
6779 ll inverse(ll a,ll MOD){
6780     return binpow(a,MOD-2,MOD);
6781 }
6782 void precompute(ll MOD){
6783     for(ll i=2;i<MAX;i++){
6784         fact[i]=(fact[i-1]*i)%MOD;
6785     }
6786     inv_fact[MAX-1]=inverse(fact[MAX-1],MOD);
6787     for(ll i=MAX-2;i>=0;i--){
6788         inv_fact[i]=(inv_fact[i+1]*(i+1))%MOD;
6789     }
6790 }
6791 ll nCr(ll a,ll b,ll MOD){
6792     if((a<0)|| (a<b)|| (b<0))
6793         return 0;
6794     ll denom=(inv_fact[b]*inv_fact[a-b])%MOD;
6795     return (denom*fact[a])%MOD;
6796 }
6797 ll power_val[MAX][MAX];
6798 void solve(){
6799     ll n,x; cin>>n>>x;
6800     ll ans=0;
6801     for(ll i=1;i<=x;i++){
6802         for(ll j=1;j<=n;j++){
6803             ll ways=(power_val[i][j]*power_val[x-i][n-j])%MOD;
6804             ways=(ways*nCr(n,j,MOD))%MOD;
6805             ans=(ans+211*min(j,n-j)*ways)%MOD;
6806         }
6807     }
6808     cout<<ans<<endl;
6809     return;
6810 }
6811 int main()

6812 {
6813     ios_base::sync_with_stdio(false);
6814     cin.tie(NULL);
6815     #ifndef ONLINE_JUDGE
6816     freopen("input.txt", "r", stdin);
6817     freopen("output.txt", "w", stdout);
6818     freopen("error.txt", "w", stderr);
6819     #endif
6820     ll test_cases=1;
6821     cin>>test_cases;
6822     precompute(MOD);
6823     for(ll i=0;i<MAX;i++){
6824         power_val[i][0]=1;
6825         for(ll j=1;j<MAX;j++){
6826             power_val[i][j]=(power_val[i][j-1]*i)%MOD;
6827         }
6828     }
6829     while(test_cases--){
6830         solve();
6831     }
6832     cout<<fixed<<setprecision(10);
6833     cerr<<"Time:"<<1000*((double)clock())/(double)CLOCKS_PER_SEC<<"ms\n";
6834 }
6835
6836 //TREESAREFUN
6837
6838 #pragma GCC optimization("O3")
6839 #pragma GCC optimization("Ofast,unroll-loops")

```

```

6840
6841 #include <bits/stdc++.h>
6842 #include <ext/pb_ds/tree_policy.hpp>
6843 #include <ext/pb_ds/assoc_container.hpp>
6844 using namespace __gnu_pbds;
6845 using namespace std;
6846 #define ll long long
6847 const ll INF_MUL=1e13;
6848 const ll INF_ADD=1e18;
6849 #define pb push_back
6850 #define mp make_pair
6851 #define nline "\n"
6852 #define f first
6853 #define s second
6854 #define pll pair<ll,ll>
6855 #define all(x) x.begin(),x.end()
6856 #define vl vector<ll>
6857 #define vvl vector<vector<ll>>
6858 #define vvvl vector<vector<vector<ll>>>
6859 #ifndef ONLINE_JUDGE
6860 #define debug(x) cerr<<"#x<<" "; _print(x); cerr<<nline;
6861 #else
6862 #define debug(x);
6863 #endif
6864 void _print(ll x){cerr<<x;}
6865 void _print(int x){cerr<<x;}
6866 void _print(char x){cerr<<x;}
6867 void _print(string x){cerr<<x;}
6868 mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());
6869 template<class T,class V> void _print(pair<T,V> p) {cerr<<"{"; _print(p.first);cerr<<" ";
; _print(p.second);cerr<<"}";}
6870 template<class T>void _print(vector<T> v) {cerr<<" [ "; for (T i:v){_print(i);cerr<<" "
; }cerr<<" ]";}
6871 template<class T>void _print(set<T> v) {cerr<<" [ "; for (T i:v){_print(i); cerr<<" ";}
cerr<<" ]";}
6872 template<class T>void _print(multiset<T> v) {cerr<<" [ "; for (T i:v){_print(i);cerr<<"
";}cerr<<" ]";}
6873 template<class T,class V>void _print(map<T, V> v) {cerr<<" [ "; for(auto i:v) {_print(i
);cerr<<" ";} cerr<<" ]";}
6874 typedef tree<ll, null_type, less<ll>, rb_tree_tag, tree_order_statistics_node_update>
ordered_set;
6875 typedef tree<ll, null_type, less_equal<ll>, rb_tree_tag,
tree_order_statistics_node_update> ordered_multiset;
6876 typedef tree<pair<ll,ll>, null_type, less<pair<ll,ll>>, rb_tree_tag,
tree_order_statistics_node_update> ordered_pset;
6877 //-----
-----
6878 const ll MOD=998244353;
6879 const ll MAX=2002000;
6880 ll n,mod;
6881 class ST{
6882 public:
6883     vector<ll> segs;
6884     ll size=0;
6885     ll ID=1;
6886
6887     ST(ll sz) {
6888         segs.assign(2*sz,ID);
6889         size=sz;
6890     }
6891
6892     ll comb(ll a,ll b) {
6893         a=(a*b)%mod;
6894         return a;
6895     }
6896
6897     void upd(ll idx, ll val) {
6898         segs[idx+=size]=val;

```

```

6899         for(idx/=2;idx;idx/=2){
6900             segs[idx]=comb(segs[2*idx],segs[2*idx+1]);
6901         }
6902     }
6903
6904     ll query(ll l,ll r) {
6905         ll lans=ID,rans=ID;
6906         for(l+=size,r+=size+1;l<r;l/=2,r/=2) {
6907             if(l&1) {
6908                 lans=comb(lans,segs[l++]);
6909             }
6910             if(r&1){
6911                 rans=comb(segs[--r],rans);
6912             }
6913         }
6914         return comb(lans,rans);
6915     }
6916 };
6917 ST dp(MAX);
6918 ll now;
6919 vector<vector<ll>> adj;
6920 vector<ll> color(MAX),freq(MAX);
6921 vector<ll> ans(MAX),tin(MAX),last(MAX);
6922 vector<ll> visited(MAX,0);
6923 void dfs(ll cur,ll par){
6924     tin[cur]=now++;
6925     dp.upd(last[color[cur]],1);
6926     dp.upd(tin[cur],freq[color[cur]]);
6927     last[color[cur]]=tin[cur];
6928     debug(cur);
6929     visited[cur]=1;
6930     for(auto chld:adj[cur]){
6931         if(visited[chld]){
6932             continue;
6933         }
6934         debug(mp(cur,chld));
6935         dfs(chld,cur);
6936     }
6937     ans[cur]=dp.query(1,tin[cur]-1);
6938 }
6939 void solve(){
6940     cin>>n>>mod;
6941     adj.clear(); adj.resize(n+5);
6942     for(ll i=1;i<=n;i++){
6943         freq[i]=0;
6944         visited[i]=0;
6945     }
6946     for(ll i=1;i<=n;i++){
6947         cin>>color[i];
6948         freq[color[i]]++;
6949     }
6950     for(ll i=1;i<n;i++){
6951         ll u,v; cin>>u>>v;
6952         adj[u].push_back(v);
6953         adj[v].push_back(u);
6954     }
6955     for(ll i=1;i<=n;i++){
6956         last[i]=i;
6957         freq[i]=max(freq[i],111);
6958         dp.upd(i,freq[i]);
6959     }
6960     now=n+1;
6961     dfs(1,-1);
6962     for(ll i=1;i<=n;i++){
6963         cout<<ans[i]<<" \n"[i==n];
6964     }
6965     return;
6966 }
6967 int main()

```

```

6968 {
6969     ios_base::sync_with_stdio(false);
6970     cin.tie(NULL);
6971     #ifndef ONLINE_JUDGE
6972     freopen("input.txt", "r", stdin);
6973     freopen("output.txt", "w", stdout);
6974     freopen("error.txt", "w", stderr);
6975     #endif
6976     ll test_cases=1;
6977     cin>>test_cases;
6978     while(test_cases--){
6979         solve();
6980     }
6981     cout<<fixed<<setprecision(10);
6982     cerr<<"Time:"<<1000*((double)clock())/((double)CLOCKS_PER_SEC)<<"ms\n";
6983 }
6984
6985 //ELEVSTRS
6986 #include <iostream>
6987 #include <fstream>
6988 #include <list>
6989 #include <stack>
6990 #include <deque>
6991 #include <utility>
6992 #include <queue>
6993 #include <set>
6994 #include <map>
6995 #include <bitset>
6996 #include <vector>
6997 #include <cmath>
6998 #include <string>
6999 #include <algorithm>
7000 #include <iomanip>
7001 #include <ctime>
7002 #include <iterator>
7003 #include <cstdio>
7004 #include <cstring>
7005 #include <cstdlib>
7006
7007
7008 using namespace std;
7009
7010 typedef long long ll;
7011 typedef long double ld;
7012 typedef unsigned long long ull;
7013
7014 #define f first
7015 #define s second
7016 #define pb push_back
7017 #define mp make_pair
7018
7019 const int maxn = 100500;
7020 const int inf = 2e9;
7021 const double eps = 1e-8;
7022 const int base = 1073676287;
7023
7024 int main()
7025 {
7026     srand( time( 0 ) );
7027     // freopen( "input.txt", "r", stdin );
7028     // freopen( "output.txt", "w", stdout );
7029     // ios_base::sync_with_stdio(false);
7030     int q;
7031     scanf ( "%d", &q );
7032     while ( q-- ) {
7033         int n, v1, v2;
7034         scanf ( "%d%d%d", &n, &v1, &v2 );

```



```

7035         if ( 2 * v1 * v1 < v2 * v2 )
7036             puts( "Elevator" );
7037         else
7038             puts( "Stairs" );
7039     }
7040     return 0;
7041 }
7042
7043 //OBTTNRNGL
7044 #include <iostream>
7045 #include <fstream>
7046 #include <list>
7047 #include <stack>
7048 #include <deque>
7049 #include <utility>
7050 #include <queue>
7051 #include <set>
7052 #include <map>
7053 #include <bitset>
7054 #include <vector>
7055 #include <cmath>
7056 #include <string>
7057 #include <algorithm>
7058 #include <iomanip>
7059 #include <ctime>
7060 #include <iterator>
7061 #include <cstdio>
7062 #include <cstring>
7063 #include <cstdlib>
7064
7065
7066 using namespace std;
7067
7068 typedef long long ll;
7069 typedef long double ld;
7070 typedef unsigned long long ull;
7071
7072 #define f first
7073 #define s second
7074 #define pb push_back
7075 #define mp make_pair
7076
7077 const int maxn = 100500;
7078 const int inf = 2e9;
7079 const double eps = 1e-8;
7080 const int base = 1073676287;
7081
7082 int main()
7083 {
7084     srand( time( 0 ) );
7085     // freopen( "input.txt", "r", stdin );
7086     // freopen( "output.txt", "w", stdout );
7087     // ios_base::sync_with_stdio(false);
7088     int q;
7089     scanf ( "%d", &q );
7090     while ( q-- ) {
7091         int n, a, b;
7092         scanf ( "%d%d%d", &n, &a, &b );
7093         if ( !( n & 1 ) && abs( a - b ) == n / 2 )
7094             puts( "0" );
7095         else
7096             printf( "%d\n", min( abs( a - b ) - 1, min( a, b ) - 1 + n - max( a, b ) ) );
7097     }
7098     return 0;
7099 }
7100
7101 //GAMSTICK
7102 #include <iostream>
7103 #include <fstream>

```

```
7104 #include <list>
7105 #include <stack>
7106 #include <deque>
7107 #include <utility>
7108 #include <queue>
7109 #include <set>
7110 #include <map>
7111 #include <bitset>
7112 #include <vector>
7113 #include <cmath>
7114 #include <string>
7115 #include <algorithm>
7116 #include <iomanip>
7117 #include <ctime>
7118 #include <iterator>
7119 #include <cstdio>
7120 #include <cstring>
7121 #include <cstdlib>
7122
7123
7124 using namespace std;
7125
7126 typedef long long ll;
7127 typedef long double ld;
7128 typedef unsigned long long ull;
7129
7130 #define f first
7131 #define s second
7132 #define pb push_back
7133 #define mp make_pair
7134
7135 const int maxn = 100500;
7136 const int inf = 2e9;
7137 const double eps = 1e-8;
7138 const int base = 1073676287;
7139
7140 string ans1;
7141 string ans2;
7142
7143 void solve() {
7144     ans1 = "Miron";
7145     ans2 = "Slava";
7146     ll n, x1, y1, x2, y2;
7147     cin >> n >> x1 >> y1 >> x2 >> y2;
7148     if ( y1 == y2 ) {
7149         cout << "Draw" << endl;
7150         return;
7151     }
7152     if ( x1 == x2 ) {
7153         if ( y1 > y2 ) {
7154             y1 = n + 1 - y1;
7155             y2 = n + 1 - y2;
7156         }
7157         ll pos = ( y1 + y2 ) / 2;
7158         if ( pos == n - pos ) {
7159             cout << "Draw" << endl;
7160             return;
7161         }
7162         if ( pos > n - pos )
7163             cout << ans1 << endl;
7164         else
7165             cout << ans2 << endl;
7166         return;
7167     }
7168
7169     if ( abs( y1 - y2 ) == 1 ) {
7170         if ( y1 > y2 ) {
7171             y1 = n + 1 - y1;
```

```

7173         y2 = n + 1 - y2;
7174     }
7175     if ( y1 > n - y1 )
7176         cout << ans1 << endl;
7177     else
7178         cout << "Draw" << endl;
7179     return;
7180 }
7181
7182 if ( y1 > y2 ) {
7183     y1 = n + 1 - y1;
7184     y2 = n + 1 - y2;
7185 }
7186 ll toGo = y2 - y1 - 2LL;
7187 y1 += ( toGo + 1 ) / 2;
7188 y2 -= toGo / 2;
7189
7190 if ( y1 > n - y1 ) {
7191     cout << ans1 << endl;
7192     return;
7193 }
7194 if ( y1 + 1 < n - y1 - 1 ) {
7195     cout << ans2 << endl;
7196     return;
7197 }
7198 cout << "Draw" << endl;
7199 // puts( "hui" );
7200 }
7201
7202 int main()
7203 {
7204     srand( time( 0 ) );
7205     // freopen( "input.txt", "r", stdin );
7206     // freopen( "output.txt", "w", stdout );
7207     ios_base::sync_with_stdio(false);
7208     int q;
7209     cin >> q;
7210     while ( q-- )
7211         solve();
7212     return 0;
7213 }
7214
7215 //MTRXMOD
7216
7217 #include <bits/stdc++.h>
7218 #include <valarray>
7219 using namespace std;
7220
7221 bool dbg = 0;
7222
7223 clock_t start_time = clock();
7224 #define current_time fixed<<setprecision(6)<<(ld) (clock()-start_time)/CLOCKS_PER_SEC
7225
7226 #define f first
7227 #define s second
7228 #define mp make_pair
7229 #define pb push_back
7230 #define all(x) (x).begin(), (x).end()
7231
7232 #define ll long long
7233 #define ld long double
7234 #define pii pair<int,int>
7235 #define umap unordered_map<int, int>
7236
7237 #define files1 freopen("input.txt","r",stdin)
7238 #define files2 freopen("out2.txt","w",stdout)
7239 #define files files1;files2
7240 #define fast_io ios_base::sync_with_stdio(0);cin.tie(0)
7241

```

```

7242 #define endl '\n'
7243 #define ln(i,n) " \n"[(i) == (n) - 1]
7244
7245 void bad(string mes = "Impossible"){cout << mes;exit(0);}
7246 void bad(int mes){cout << mes;exit(0);}
7247
7248 template<typename T>
7249 string bin(T x, int st = 2){
7250     string ans = "";
7251     while (x > 0){
7252         ans += char('0' + x % st);
7253         x /= st;
7254     }
7255     reverse(ans.begin(), ans.end());
7256     return ans.empty() ? "0" : ans;
7257 }
7258
7259 template<typename T>
7260 void amax(T& x, T y) {
7261     x = max(x, y);
7262 }
7263
7264 template<typename T>
7265 void amin(T& x, T y) {
7266     x = min(x, y);
7267 }
7268
7269 inline int popcount(int x){
7270     int count = 0;
7271     __asm__ volatile("POPCNT %1, %0;":"=r"(count):"r"(x));
7272     return count;
7273 }
7274
7275 template<typename T>
7276 T input(){
7277     T ans = 0, m = 1;
7278     char c = ' ';
7279
7280     while (!(c >= '0' && c <= '9') || c == '-')) {
7281         c = getchar();
7282     }
7283
7284     if (c == '-')
7285         m = -1, c = getchar();
7286     while (c >= '0' && c <= '9'){
7287         ans = ans * 10 + (c - '0'), c = getchar();
7288     }
7289     return ans * m;
7290 }
7291
7292 template<typename T> void read(T& a) { a = input<T>(); }
7293 template<typename T> void read(T& a, T& b) { read(a), read(b); }
7294 template<typename T> void read(T& a, T& b, T& c) { read(a, b), read(c); }
7295 template<typename T> void read(T& a, T& b, T& c, T& d) { read(a, b), read(c, d); }
7296
7297 const int inf = 1e9 + 20;
7298 const short short_inf = 3e4 + 20;
7299 const long double eps = 1e-12;
7300 const int maxn = 1e5 + 12, base = 1e9 + 7;
7301 const ll llinf = 2e18 + 5;
7302
7303 template<typename T>
7304 T bnpow(T n, T s)
7305 {
7306     if (s <= 0)
7307         return 1LL;
7308     if (s % 2 == 0){
7309         T b = bnpow(n, s / 2);
7310         return ( 1LL * b * b ) % base;

```

```

7311     } else {
7312         return (1LL* binpow(n, s - 1) * n) % base;
7313     }
7314 }
7315
7316 int b[1011][1011];
7317 int c[1011], d[1011], a[1001], n;
7318
7319 void print() {
7320     int x = a[0];
7321     for (int i = 0; i < n; i++) {
7322         a[i] -= x;
7323         c[i] = -a[i];
7324     }
7325     bool le = 1;
7326     for (int i = 0; i < n; i++) {
7327         if (a[i] < c[i]) {
7328             le = 1;
7329             break;
7330         }
7331         if (a[i] > c[i]) {
7332             le = 0;
7333             break;
7334         }
7335     }
7336     if (le) {
7337         for (int i = 0; i < n; i++) {
7338             printf("%d", a[i]);
7339             if (i != n - 1)
7340                 printf(" ");
7341         }
7342         puts("");
7343     } else {
7344         for (int i = 0; i < n; i++) {
7345             printf("%d", c[i]);
7346             if (i != n - 1)
7347                 printf(" ");
7348         }
7349         puts("");
7350     }
7351 }
7352
7353 int main() {
7354     int q;
7355     // files1;
7356     read(n, q);
7357
7358     pii res = mp(-inf, 0);
7359     for (int i = 0; i < n; i++)
7360         for (int j = 0; j < n; j++) {
7361             read(b[i][j]);
7362             amax(res, mp(b[i][j], i));
7363         }
7364
7365     int r = res.s;
7366     for (int i = 0; i < n; i++) {
7367         a[i] = b[r][i];
7368     }
7369
7370     print();
7371
7372     for (int i = 0; i < q; i++) {
7373         int row;
7374         read(row);
7375         row--;
7376         for (int j = 0; j < n; j++) {
7377             read(d[j]);
7378
7379             b[row][j] = d[j];

```

```

7380         b[j][row] = d[j];
7381     }
7382
7383     if (row == 0) {
7384         a[0] = a[1] - b[1][0];
7385         bool ok = 1;
7386         for (int j = 0; j < n; j++) {
7387             if (b[0][j] != abs(a[0] - a[j])) {
7388                 ok = 0;
7389                 break;
7390             }
7391         }
7392
7393         if (!ok) {
7394             a[0] = a[1] + b[1][0];
7395
7396             ok = 1;
7397             for (int j = 0; j < n; j++) {
7398                 if (b[0][j] != abs(a[0] - a[j]))
7399                     assert(0);
7400             }
7401         }
7402     } else {
7403         a[row] = a[0] - b[0][row];
7404         bool ok = 1;
7405         for (int j = 0; j < n; j++) {
7406             if (b[row][j] != abs(a[row] - a[j])) {
7407                 ok = 0;
7408                 break;
7409             }
7410         }
7411
7412         if (!ok) {
7413             a[row] = a[0] + b[0][row];
7414
7415             ok = 1;
7416             for (int j = 0; j < n; j++) {
7417                 if (b[row][j] != abs(a[row] - a[j]))
7418                     assert(0);
7419             }
7420         }
7421     }
7422
7423     print();
7424 }
7425 return 0;
7426 }
7427
7428 //TRCNTCT
7429 #include <iostream>
7430 #include <fstream>
7431 #include <list>
7432 #include <stack>
7433 #include <deque>
7434 #include <utility>
7435 #include <queue>
7436 #include <set>
7437 #include <map>
7438 #include <bitset>
7439 #include <vector>
7440 #include <cmath>
7441 #include <string>
7442 #include <algorithm>
7443 #include <iomanip>
7444 #include <ctime>
7445 #include <iterator>
7446 #include <cstdio>
7447 #include <cstring>
7448 #include <cstdlib>

```

```

7449
7450
7451 using namespace std;
7452
7453 typedef long long ll;
7454 typedef long double ld;
7455 typedef unsigned long long ull;
7456
7457 #define f first
7458 #define s second
7459 #define pb push_back
7460 #define mp make_pair
7461
7462 const int maxn = 1000500;
7463 const int inf = 2e9;
7464 const double eps = 1e-8;
7465 const int base = 1073676287;
7466
7467 vector < int > edge[maxn];
7468 int used[maxn];
7469 // int timer;
7470 // int anc[maxn];
7471 // int SZ[maxn];
7472
7473 // void dfsBrute( int v, int l, int r ) {
7474 //     used[v] = true;
7475 //     int sz = edge[v].size();
7476 //     for ( int j = 0; j < sz; j++ ) {
7477 //         int to = edge[v][j];
7478 //         if ( used[to] )
7479 //             continue;
7480 //         if ( to > r || to < l )
7481 //             continue;
7482 //         dfsBrute( to, l, r );
7483 //     }
7484 // }
7485
7486 // bool correct( int n, int l, int r ) {
7487 //     timer = 0;
7488 //     for ( int j = 1; j <= r; j++ )
7489 //         used[j] = 0;
7490 //     for ( int j = 1; j <= r; j++ )
7491 //         if ( !used[j] ) {
7492 //             ++timer;
7493 //             if ( timer == 2 )
7494 //                 return false;
7495 //             dfsBrute( j, l, r );
7496 //         }
7497 //     return true;
7498 // }
7499
7500 // ll brute( int n ) {
7501 //     ll ans = 0;
7502 //     for ( int j = 1; j <= n; j++ )
7503 //         for ( int i = 1; i <= j; i++ )
7504 //             ans += correct( n, i, j );
7505 //     return ans;
7506 // }
7507
7508 // int findSet( int v ) {
7509 //     return v == anc[v] ? v : anc[v] = findSet( anc[v] );
7510 // }
7511
7512 // void uniteSets( int u, int v ) {
7513 //     u = findSet( u );
7514 //     v = findSet( v );
7515 //     if ( SZ[u] < SZ[v] )
7516 //         swap( u, v );
7517 //     SZ[u] += SZ[v];

```

```

7518 // anc[v] = u;
7519 // }
7520
7521 ll ans;
7522 int low[maxn];
7523 int high[maxn];
7524 int prefSum[maxn];
7525
7526 void dfs( int v, int anc, int l, int r ) {
7527     for ( int to : edge[v] ) {
7528         if ( used[to] )
7529             continue;
7530         if ( to == anc )
7531             continue;
7532         if ( to < l || to > r )
7533             continue;
7534         low[to] = min( to, low[v] );
7535         high[to] = max( to, high[v] );
7536         dfs( to, v, l, r );
7537     }
7538 }
7539
7540 inline int getSum( int l, int r ) {
7541     return l > r ? 0 : prefSum[r] - prefSum[l - 1];
7542 }
7543
7544 inline void solution( int l, int r ) {
7545     if ( l > r )
7546         return;
7547     int center = ( l + r ) >> 1;
7548     for ( int j = l; j <= r; j++ ) {
7549         low[j] = -inf;
7550         high[j] = inf;
7551     }
7552     low[center] = high[center] = center;
7553     dfs( center, -1, l, r );
7554     for ( int j = center - 1; j >= l; j-- ) {
7555         low[j] = min( low[j], low[j + 1] );
7556         high[j] = max( high[j], high[j + 1] );
7557     }
7558     for ( int j = center + 1; j <= r; j++ ) {
7559         low[j] = min( low[j], low[j - 1] );
7560         high[j] = max( high[j], high[j - 1] );
7561     }
7562     prefSum[center - 1] = 0;
7563     prefSum[center] = 1;
7564     for ( int j = center + 1; j <= r; j++ )
7565         prefSum[j] = prefSum[j - 1] + ( high[j] == j );
7566
7567     int R = center;
7568     for ( int j = center; j >= l; j-- ) {
7569         if ( low[j] != j )
7570             continue;
7571         while ( R <= r && low[R] >= j )
7572             ++R;
7573         ans += 1LL * getSum( high[j], R - 1 );
7574     }
7575     used[center] = true;
7576     if ( l != r ) {
7577         solution( l, center - 1 );
7578         solution( center + 1, r );
7579     }
7580 }
7581
7582 int main()
7583 {
7584     srand( time( NULL ) );
7585     // freopen( "input.txt", "r", stdin );
7586     // freopen( "output.txt", "w", stdout );

```



```

7587 // ios_base::sync_with_stdio(false);
7588
7589
7590 int q;
7591 scanf ( "%d", &q );
7592 while ( q-- ) {
7593     int n;
7594     scanf ( "%d", &n );
7595     for ( int j = 1; j < n; j++ ) {
7596         int u, v;
7597         scanf ( "%d%d", &u, &v );
7598         edge[u].pb( v );
7599         edge[v].pb( u );
7600     }
7601     for ( int j = 1; j <= n; j++ )
7602         used[j] = false;
7603     ans = 0LL;
7604     solution( 1, n );
7605     for ( int j = 1; j <= n; j++ )
7606         edge[j].clear();
7607     cout << ans << endl;
7608 }
7609
7610
7611 // int it = 50000;
7612 // int maxN = 10;
7613 // for ( int qq = 1; qq <= it; qq++ ) {
7614 //     int n = 1 + rand() % maxN;
7615 //     for ( int j = 1; j <= n; j++ ) {
7616 //         edge[j].clear();
7617 //         anc[j] = j;
7618 //         SZ[j] = 1;
7619 //         used[j] = false;
7620 //     }
7621 //     for ( int j = 1; j < n; j++ ) {
7622 //         int u = 1 + rand() % n;
7623 //         int v = 1 + rand() % n;
7624 //         while ( findSet( u ) == findSet( v ) ) {
7625 //             u = 1 + rand() % n;
7626 //             v = 1 + rand() % n;
7627 //         }
7628 //         edge[u].pb( v );
7629 //         edge[v].pb( u );
7630 //         uniteSets( u, v );
7631 //     }
7632 //     ans = 0;
7633 //     solution( 1, n );
7634
7635 //     if ( brute( n ) != ans ) {
7636 //         puts( "kek" );
7637 //         printf( "%d\n", n );
7638 //         for ( int j = 1; j <= n; j++ )
7639 //             for ( int i : edge[j] )
7640 //                 if ( j < i )
7641 //                     printf( "%d %d\n", j, i );
7642 //         printf( "\n" );
7643 //         cout << brute( n ) << ' ' << ans << endl;
7644 //         return 0;
7645 //     }
7646 //     // if ( qq % ( it / 100 ) == 0 )
7647 //     //     printf( "%d\n", qq / ( it / 100 ) );
7648 // }
7649 // puts( "all good" );
7650 return 0;
7651 }
7652
7653 //LCM3GCD2
7654 /**
7655 * Author : RDP

```

```

7656     * There are no two words in the English language more harmful than "good job".
7657     * 1729 ;)
7658     **/
7659 #include <bits/stdc++.h>
7660
7661 using namespace std;
7662 using ll = long long;
7663
7664 /***** Definations, Macros and Debug Stuff *****/
7665 void debug_out() { cerr << '\n'; }
7666 string to_string(const string &s) { return s; }
7667 template <typename Head, typename... Tail>
7668 void debug_out(Head H, Tail... T)
7669 {
7670     cerr << " " << to_string(H);
7671     debug_out(T...);
7672 }
7673
7674 #define endl '\n'
7675 #define debug(...) cerr << "[" << #__VA_ARGS__ << "]: ", debug_out(__VA_ARGS__)
7676 #define GODSPEED \
7677     ios::sync_with_stdio(false); \
7678     std::cin.tie(NULL); \
7679     std::cout.tie(NULL);
7680 #define all(x) (x).begin(), (x).end()
7681 const long double EPS = 5e-8;
7682 #define PI 3.1415926535897932384626433832795
7683 const ll MOD = 1000000007;
7684 /***** Frequently used functions *****/
7685 template <typename T>
7686 inline void print_vector(vector<T> &a)
7687 {
7688     for (auto &x : a)
7689         cout << x << ' ';
7690     cout << endl;
7691 }
7692
7693 inline ll binary_pow(ll a, ll b)
7694 {
7695     ll res = 1;
7696     while (b > 0)
7697     {
7698         if (b & 1)
7699             res = res * a;
7700         a = a * a;
7701         b >>= 1;
7702     }
7703     return res;
7704 }
7705 inline ll mod_pow(ll x, ll y, ll m = MOD)
7706 {
7707     ll res = 1;
7708     x = x % m;
7709     if (x == 0)
7710         return 0;
7711     while (y > 0)
7712     {
7713         if (y & 1)
7714             res = (res * x) % m;
7715         y = y >> 1;
7716         x = (x * x) % m;
7717     }
7718     return res;
7719 }
7720 inline ll mod_add(ll a, ll b, ll m = MOD)
7721 {
7722     a = a % m;
7723     b = b % m;
7724     return ((a + b) % m + m) % m;

```

```

7725 }
7726 inline ll mod_mul(ll a, ll b, ll m = MOD)
7727 {
7728     a = a % m;
7729     b = b % m;
7730     return ((a * b) % m) + m) % m;
7731 }
7732 inline ll mod_sub(ll a, ll b, ll m = MOD)
7733 {
7734     a = a % m;
7735     b = b % m;
7736     return (((a - b) % m) + m) % m;
7737 }
7738 inline ll mminvprime(ll a, ll b)
7739 {
7740     return mod_pow(a, b - 2, b);
7741 }
7742 inline ll mod_div(ll a, ll b, ll m = MOD)
7743 {
7744     a = a % m;
7745     b = b % m;
7746     return (mod_mul(a, mminvprime(b, m), m) + m) % m;
7747 }
7748 inline ll ceilf(ll x, ll y)
7749 {
7750     return x % y == 0 ? x / y : x / y + 1;
7751 }
7752 // Use this for randomizing things
7753 mt19937_64 rng(chrono::steady_clock::now().time_since_epoch().count());
7754
7755 set<ll> primes;
7756 vector<bool> is_prime;
7757 void precompute_primes(ll n)
7758 {
7759     is_prime.resize(n + 1, 1);
7760     for (ll p = 2; p * p <= n; p++)
7761     {
7762         if (is_prime[p] == true)
7763         {
7764             for (ll i = p * p; i <= n; i += p)
7765                 is_prime[i] = false;
7766         }
7767     }
7768     for (ll p = 2; p <= n; p++)
7769         if (is_prime[p])
7770             primes.insert(p);
7771 }
7772
7773 /*****/
7774 map<pair<int, int>, ll> gcd_cache;
7775 map<tuple<int, int, int>, ll> lcm_cache;
7776 int numq = 0;
7777
7778 class FracInt
7779 {
7780 public:
7781     map<ll, ll> up, down;
7782     FracInt() { ; }
7783     FracInt(__int128_t x)
7784     {
7785         for (auto p : primes)
7786         {
7787             if (p > x)
7788                 break;
7789             while (x % p == 0)
7790                 x /= p, up[p]++;
7791             up[1]++;
7792         }
7793     }

```

```

7794 void mul(__int128_t x)
7795 {
7796     for (auto p : primes)
7797     {
7798         if (p > x)
7799             break;
7800         while (x % p == 0)
7801             x /= p, up[p]++;
7802     }
7803 }
7804 void div(__int128_t x)
7805 {
7806     for (auto p : primes)
7807     {
7808         if (p > x)
7809             break;
7810         while (x % p == 0)
7811             x /= p, down[p]++;
7812     }
7813     normalize();
7814 }
7815 void mul(FracInt y)
7816 {
7817     for (auto p : y.up)
7818     {
7819         up[p.first] += p.second;
7820     }
7821     for (auto p : y.down)
7822     {
7823         down[p.first] += p.second;
7824     }
7825     normalize();
7826 }
7827 void div(FracInt y)
7828 {
7829     for (auto p : y.up)
7830     {
7831         down[p.first] += p.second;
7832     }
7833     for (auto p : y.down)
7834     {
7835         up[p.first] += p.second;
7836     }
7837     normalize();
7838 }
7839 void normalize()
7840 {
7841     for (auto &p : up)
7842     {
7843         p.second -= down[p.first];
7844         down[p.first] = 0;
7845     }
7846 }
7847 __int128_t get_abs()
7848 {
7849     normalize();
7850     __int128_t val = 1;
7851     for (auto &p : up)
7852     {
7853         for (int i = 0; i < p.second; i++)
7854             val *= p.first;
7855     }
7856     return val;
7857 }
7858 };
7859
7860 FracInt cuberoot(FracInt x)
7861 {
7862     x.normalize();

```

```

7863     for (auto &p : x.up)
7864         p.second /= 3;
7865     return x;
7866 }
7867
7868 ll query(int i, int j, int k = -1)
7869 {
7870     i++;
7871     j++;
7872     if (k != -1)
7873         k++;
7874     if (k == -1)
7875     {
7876         if (gcd_cache.count({i, j}))
7877             return gcd_cache[{i, j}];
7878     }
7879     else
7880     {
7881         if (lcm_cache.count({i, j, k}))
7882             return lcm_cache[{i, j, k}];
7883     }
7884     numq++;
7885     if (k == -1)
7886         cout << "1 " << i << " " << j << endl;
7887     else
7888         cout << "2 " << i << " " << j << " " << k << endl;
7889     ll x;
7890     cin >> x;
7891     if (k == -1)
7892     {
7893         gcd_cache[{i, j}] = x;
7894         gcd_cache[{j, i}] = x;
7895     }
7896     else
7897     {
7898         vector<int> tmp = {i, j, k};
7899         sort(all(tmp));
7900         do
7901         {
7902             lcm_cache[{tmp[0], tmp[1], tmp[2]}] = x;
7903         } while (next_permutation(all(tmp)));
7904     }
7905     assert(x != -1);
7906     return x;
7907 }
7908 void terminate(vector<ll> a)
7909 {
7910     cout << "3 ";
7911     print_vector(a);
7912     ll x;
7913     cin >> x;
7914     assert(x == 1);
7915 }
7916 FracInt solve_for_mul(int i, int j, int k)
7917 {
7918     // let a,b,c
7919     __int128_t lcmabc = (query(i, j, k));
7920
7921     __int128_t gcdab = (query(i, j));
7922     __int128_t gcdac = (query(i, k));
7923     __int128_t gcdbc = (query(j, k));
7924     __int128_t gcdabc = __int128_t(gcd(ll(gcdab), ll(gcdbc)));
7925     FracInt abc(1);
7926     abc.mul(lcmabc);
7927     abc.mul(gcdab);
7928     abc.mul(gcdac);
7929     abc.mul(gcdbc);
7930     abc.div(gcdabc);
7931

```

```

7932     return abc;
7933 }
7934 vector<ll> solve_for4(int a, int b, int c, int d)
7935 {
7936     // let a,b,c,d be first 4 elements.
7937     FracInt abc = solve_for_mul(a, b, c);
7938     FracInt bcd = solve_for_mul(b, c, d);
7939     FracInt acd = solve_for_mul(a, c, d);
7940     FracInt abd = solve_for_mul(a, b, d);
7941
7942     FracInt abcd3(1);
7943     abcd3.mul(abc);
7944     abcd3.mul(bcd);
7945     abcd3.mul(acd);
7946     abcd3.mul(abd);
7947
7948     vector<ll> ans(4);
7949     FracInt a0 = cuberoot(abcd3);
7950     a0.div(bcd);
7951
7952     ans[0] = ll(a0.get_abs());
7953
7954     FracInt a1 = cuberoot(abcd3);
7955     a1.div(acd);
7956
7957     ans[1] = ll(a1.get_abs());
7958
7959     FracInt a2 = cuberoot(abcd3);
7960     a2.div(abd);
7961
7962     ans[2] = ll(a2.get_abs());
7963
7964     FracInt a3 = cuberoot(abcd3);
7965     a3.div(abc);
7966
7967     ans[3] = ll(a3.get_abs());
7968     return ans;
7969 }
7970 void solve_for_index(vector<ll> &A)
7971 {
7972     // we want c from a,b,c
7973     __int128_t a = A[A.size() - 2];
7974     __int128_t b = A[A.size() - 1];
7975     int i = A.size() - 2, j = A.size() - 1, k = A.size();
7976     __int128_t lcmabc = query(i, j, k);
7977     __int128_t gcdab = query(i, j);
7978     __int128_t gcdac = query(i, k);
7979     __int128_t gcdbc = query(j, k);
7980     __int128_t gcdabc = gcd(ll(gcdab), ll(gcdbc));
7981
7982     ll c = ll((lcmabc * gcdab * gcdac * gcdbc) / (a * b * gcdabc));
7983     A.push_back(c);
7984     return;
7985 }
7986
7987 void test_case()
7988 {
7989     numq = 0;
7990     ll t = 1e6;
7991     gcd_cache.clear();
7992     lcm_cache.clear();
7993     int n;
7994     cin >> n;
7995     auto a = solve_for4(0, 1, 2, 3);
7996     for (int i = 4; i < n; i++)
7997     {
7998         solve_for_index(a);
7999     }
8000     terminate(a);

```

```

8001     return;
8002 }
8003
8004 int main()
8005 {
8006     // GODSPEED;
8007     precompute_primes(1e6 + 10);
8008     int t = 1;
8009     cin >> t;
8010     for (int i = 1; i <= t; i++)
8011     {
8012         // cout << "Case #" << i << ": ";
8013         test_case();
8014     }
8015     return 0;
8016 }
8017 /*
8018 Some things to remember when you're stuck:
8019     1. Check for edge cases.
8020     2. Stay Calm.
8021     3. Don't be stupid (search for silly mistakes).
8022     4. Read problem again. Approach solution from different point of view.
8023     5. In case of modulo, check for negative result (add MOD).
8024
8025 Some common C++ pit falls:
8026     1. Don't use inbuilt ceil.
8027     2. Never take inputs as double unless it is necessary.
8028     3. Don't pass INT in accumulate.
8029 */
8030
8031 //MEX_SEQ
8032 #include<bits/stdc++.h>
8033 #include<ext/pb_ds/assoc_container.hpp>
8034 #include<ext/pb_ds/tree_policy.hpp>
8035 #pragma GCC optimize "trapv"
8036 #define F first
8037 #define S second
8038 // #define endl "\n"
8039 #define Endl "\n"
8040 #define fbo find_by_order
8041 #define ook order_of_key
8042 #define ll long long
8043 #define ld long double
8044 #define vl vector<long long>
8045 #define pll pair<long long,long long>
8046 #define sl set<long long>
8047 #define uset unordered_set
8048 #define umap unordered_map
8049 #define prq priority_queue
8050 #define pqll priority_queue<ll>
8051 #define pb push_back
8052 #define ppb pop_back
8053 #define mp make_pair
8054 #define bpc(x) __builtin_popcount(x)
8055 #define sz(v) (int)(v.size())
8056 #define all(v) (v).begin(),(v).end()
8057 #define mem(a, val) memset(a, val, sizeof(a))
8058 #define mem0(a) memset(a,0,sizeof(a))
8059 #define mem1(a) memset(a,-1,sizeof(a))
8060 #define N 1000000
8061 #define N2 2000000
8062
8063 const long double EPS = 0.0001;
8064 const long double PI = 3.141592653589793238;
8065 const long long hell = 1000000007;
8066 const long long mod = 998244353;
8067 const long long INF = 1e16;
8068 using namespace std;
8069 using namespace __gnu_pbds;

```

```

8070     typedef tree<ll, null_type, less<ll>, rb_tree_tag,tree_order_statistics_node_update>
ordered_set;
8071     mt19937 rng ((unsigned int) chrono::steady_clock::now().time_since_epoch().count());
8072
8073     template<typename T, typename U> static inline void amin(T &x, U y){ if(y < x) x = y; }
8074     template<typename T, typename U> static inline void amax(T &x, U y){ if(x < y) x = y; }
8075     ll power(ll x, ll y, ll p=hell)
8076     {
8077         ll res = 1;
8078         x = x % p;
8079         while (y > 0)
8080         {
8081             if (y & 1)
8082                 res = (res*x) % p;
8083             y = y>>1; // y = y/2
8084             x = (x*x) % p;
8085         }
8086         return res;
8087     }
8088
8089     // Returns n^(-1) mod p
8090     ll modInverse(ll n, ll p=hell)
8091     {
8092         return power(n, p-2, p);
8093     }
8094
8095     // Returns nCr % p using Fermat's little theorem.
8096     ll fac[N+1];
8097     ll power2[N+1];
8098     ll mInv[N+1];
8099     ll facInv[N+1];
8100     void pre(ll p=hell){
8101         fac[0] = 1;
8102         power2[0]=1;
8103         mInv[0]=1;
8104         mInv[1]=1;
8105         facInv[0]=1;
8106         facInv[1]=1;
8107         for (ll i=1 ; i<=N; i++) {
8108             fac[i] = (fac[i-1]*i)%p;
8109             power2[i] = (power2[i-1]*2)%p;
8110             if(i>1){
8111                 mInv[i]=(mInv[p%i]*(p-p/i))%p;
8112                 facInv[i]=(facInv[i-1]*mInv[i])%p;
8113             }
8114         }
8115     }
8116     ll nCrModPFermat(ll n, ll r, ll p=hell)
8117     {
8118         if(r>n)
8119             return 0;
8120         if (r==0)
8121             return 1;
8122
8123         return (fac[n]* facInv[r] % p * facInv[n-r] % p) % p;
8124         // return (fac[n]* modInverse(fac[r],p) % p * modInverse(fac[n-r],p) % p) % p;
8125     }
8126     void solve(ll n,ll m){
8127
8128         m=min(n,m);
8129         if(m==0){
8130             cout<<1<<<endl;
8131             return;
8132         }
8133         ll s=1;
8134         ll ans=0;
8135
8136         if(m>1){
8137             ans=s*power2[n-2];

```



```

8138         for (ll i=2;i<=n-1;i++){
8139             // cout<<ans<<" ";
8140             ll temp=0;
8141             if(m>2){
8142                 temp=(2*(s-1-nCrModPFermat(i-2,m-2)+hell))%hell;
8143                 temp=(temp+1+nCrModPFermat(i-1,m-2))%hell;
8144             }
8145             else{
8146                 temp=1;
8147             }
8148             s=temp;
8149             ans=(ans+(temp*power2[n-i-1])%hell)%hell;
8150         }
8151     }
8152     // cout<<ans;
8153     for (ll i=0;i<=m;i++){
8154         ans=(ans+nCrModPFermat(n-1,i))%hell;
8155     }
8156     cout<<(ans+1)%hell<<endl;
8157 }
8158 int main(){
8159
8160     ios_base::sync_with_stdio(0);
8161     cin.tie(NULL);
8162     cout.tie(NULL);
8163     #ifndef ONLINE_JUDGE
8164         freopen("tests/output_10.in", "r", stdin);
8165         freopen("tests/output_10.out", "w", stdout);
8166     #endif
8167
8168     ll t,n,m;
8169     cin>>t;
8170     pre();
8171     ll sum_n=0;
8172     ll sum_m=0;
8173     while(t--){
8174         cin>>n>>m;
8175         sum_n+=n;
8176         sum_m+=m;
8177         solve(n,m);
8178     }
8179     cerr<<sum_n<<" "<<sum_m;
8180     assert(sum_m<=N2);
8181     assert(sum_n<=N2);
8182 }
8183
8184 //AWESUM_OR
8185 #include<bits/stdc++.h>
8186 using namespace std;
8187 #define int long long
8188
8189 const int mod=1e9+7;
8190 vector<int> pcom(61, 0);
8191
8192 int binexp(int a, int b, int mod){
8193     assert(b>=0);
8194     a=a%mod;
8195     int ans = 1;
8196     while(b){
8197         if(b&1){
8198             ans=ans*a%mod;
8199         }
8200         a=a*a%mod;
8201         b/=2;
8202     }
8203     return ans;
8204 }
8205
8206 }

```

```

8207
8208 void solve(){
8209     int n;
8210     cin>>n;
8211
8212     int x = __builtin_popcountll(n);
8213     cout<<pcom[x]*6%mod<<'\\n';
8214     // cout<<((binexp(3, x, mod)-3*binexp(2, x, mod)%mod+mod)%mod+3)%mod<<'\\n';
8215 }
8216
8217 signed main(){
8218
8219     ios::sync_with_stdio(false);
8220     cin.tie(0); cout.tie(0);
8221
8222     for(int a=3; a<=60; a++){
8223         int sum = 0;
8224         for(int b = a-1; b>0; b--){
8225             for(int c = b-1; c>0; c--){
8226                 sum = (sum + binexp(2, b-c-1, mod)*binexp(3, c-1, mod)%mod)%mod;
8227             }
8228         }
8229         pcom[a] = sum;
8230     }
8231
8232     int tt;
8233     cin>>tt;
8234
8235     while(tt-->0) solve();
8236 }
8237
8238 //DIGITOP
8239 #include<bits/stdc++.h>
8240 using namespace std;
8241
8242 #define mod 1000000007
8243 typedef set<string> ss;
8244 typedef vector<int> vs;
8245 typedef map<int, char> msi;
8246 typedef pair<int, int> pa;
8247 typedef long long int ll;
8248
8249 ll n, k, i, j, val, ca[10], cb[10];
8250 string a[100005], b[100005];
8251 int main()
8252 {
8253     ios_base::sync_with_stdio(false);
8254     cin.tie(0);
8255 #ifndef ONLINE_JUDGE
8256     freopen("inputf.in", "r", stdin);
8257     freopen("output.txt", "w", stdout);
8258 #endif
8259
8260     int t;
8261     cin >> t;
8262     while (t-->0)
8263     {
8264         memset(ca, 0, sizeof(ca));
8265         memset(cb, 0, sizeof(cb));
8266         cin >> n >> k;
8267         for (i = 0; i < n; i++)
8268             cin >> a[i];
8269         for (i = 0; i < n; i++)
8270             cin >> b[i];
8271         for (i = 0; i < n; i++)
8272             if (a[i].length() != b[i].length())
8273                 break;
8274         if (i != n)
8275             cout << "NO\\n";

```

```

8276     else
8277     {
8278         for (i = 0; i < n; i++)
8279         {
8280             for (j = 0; j < a[i].length(); j++)
8281                 ca[a[i][j] - '0']++;
8282             for (j = 0; j < b[i].length(); j++)
8283                 cb[b[i][j] - '0']++;
8284         }
8285         val = 0;
8286         for (i = 0; i < 10; i++)
8287             val += max(0LL, cb[i] - ca[i]);
8288         if (val <= k)
8289             cout << "YES\n";
8290         else
8291             cout << "NO\n";
8292     }
8293 }
8294
8295 return 0;
8296 }
8297
8298 //ROTMIN
8299 #include "bits/stdc++.h"
8300 using namespace std;
8301
8302 typedef long long        lol;
8303 typedef std::pair<int,int> pii;
8304 #define pb                push_back
8305 #define ub                upper_bound
8306 #define lb                lower_bound
8307 #define fo(i,l,r,d)      for (auto i=(l); (d)<0?i>(r):((d)>0?i<(r):0); i+=(d))
8308 #define all(x)            x.begin(), x.end()
8309 #define ff                first
8310 #define ss                second
8311
8312 std::mt19937 rng (std::chrono::high_resolution_clock::now().time_since_epoch().count());
8313 template <typename A, typename B> std::ostream& operator<< (std::ostream &cout, const
8314 std::pair<A, B> &p) { return cout << p.first << ' ' << p.second; } template <typename A,
size_t n> std::ostream& operator<< (std::ostream &cout, const std::array<A, n> &v) {
for (int i = 0; i < n - 1; ++i) cout << v[i] << ' '; return (n ? cout << v.back(): cout
<< '\n'); } template <typename A> std::ostream& operator<< (std::ostream &cout, const
std::vector<A> &v) { for (int i = 0; i < v.size() - 1; ++i) cout << v[i] << ' '; return
(v.size() ? cout << v.back(): cout << '\n'); }
8315 template <typename A, typename B> std::istream& operator>> (std::istream &cin, std::pair
<A, B> &p) { cin >> p.first; return cin >> p.second; } template <typename A, size_t n>
std::istream& operator>> (std::istream &cin, std::array<A, n> &v) { assert(n); for (int
i = 0; i < n - 1; i++) cin >> v[i]; return cin >> v.back(); } template <typename A> std
::istream& operator>> (std::istream &cin, std::vector<A> &v) { assert(v.size()); for (
int i = 0; i < v.size() - 1; i++) cin >> v[i]; return cin >> v.back(); }
8316 template <typename A, typename B> auto amax (A &a, const B b){ if (b > a) a = b ; return
a; }
8317 template <typename A, typename B> auto amin (A &a, const B b){ if (b < a) a = b ; return
a; }
8318
8319
8320
8321 void darling (const int kase) {
8322
8323     int n, p, q; string a;
8324     cin >> n >> p >> q >> a;
8325
8326     int l = 0, r = n + 1;
8327
8328     while (l < r - 1) {
8329         int m = (l + r) / 2;
8330
8331         vector<pii> op;

```

```

8332
8333     for (int i = 0; i < m; i++)
8334         op.pb(pii(('z' - a[i]) + 1, a[i] - 'a'));
8335
8336     sort(all(op));
8337
8338     int suf_dn = 0, pre_up = 0;
8339     for (auto [up, dn]: op)
8340         suf_dn += dn;
8341
8342     int pbl = (suf_dn <= q);
8343     for (auto [up, dn]: op)
8344         pre_up += up, suf_dn -= dn,
8345         pbl |= (pre_up <= p and suf_dn <= q);
8346
8347     if (pbl) l = m;
8348     else r = m;
8349 }
8350
8351 if (l == n)
8352     return void(cout << string(n, 'a') << '\n');
8353
8354
8355 vector<pii> op;
8356
8357 for (int i = 0; i < l; i++)
8358     op.pb(pii(('z' - a[i]) + 1, a[i] - 'a'));
8359
8360 sort(all(op));
8361
8362 int suf_dn = 0, pre_up = 0;
8363 for (auto [up, dn]: op)
8364     suf_dn += dn;
8365
8366 int max_dn_left = (suf_dn <= q ? q - suf_dn: 0);
8367 int up_left = (suf_dn <= q ? p: 0);
8368
8369 for (auto [up, dn]: op) {
8370     pre_up += up, suf_dn -= dn;
8371     if (pre_up <= p and suf_dn <= q)
8372         max_dn_left = q - suf_dn,
8373         up_left = p - pre_up;
8374 }
8375
8376 fo(i, 0, l, 1)
8377     a[i] = 'a';
8378
8379 a[l] -= max_dn_left;
8380
8381 fo(i, l+1, n, 1)
8382     if (a[i] + up_left > 'z')
8383         up_left -= 'z' - a[i] + 1,
8384         a[i] = 'a';
8385
8386 cout << a << '\n';
8387
8388 }
8389
8390 int main () {
8391     ios_base::sync_with_stdio(0), cin.tie(0);
8392
8393     int t; cin >> t, assert(t >= 0);
8394     for (int i = 0; t--; )
8395         darling(++i);
8396
8397 }
8398
8399 //XOR_ORDER
8400 #include "bits/stdc++.h"

```

```

8401 #include <ext/pb_ds/assoc_container.hpp>
8402 #include <ext/pb_ds/tree_policy.hpp>
8403
8404 using namespace std;
8405 using namespace __gnu_pbds;
8406
8407 #define all(x)      x.begin(), x.end()
8408 #define pb          push_back
8409 #define sz(x)       (int)(x.size())
8410 #define ll          long long
8411 #define fi          first
8412 #define se          second
8413 #define lbd         lower_bound
8414 #define ubd         upper_bound
8415
8416 template <typename T>
8417 using ordered_set = tree<T, null_type,
8418     less<T>, rb_tree_tag,
8419     tree_order_statistics_node_update>;
8420
8421 const int MOD = 1e9 + 7;
8422 const double eps = 1e-10;
8423 const long long INF = 1e18;
8424 const int N = 2e5 + 10;
8425
8426 void solve() {
8427     int a, b, c;
8428     cin >> a >> b >> c;
8429
8430     int ans = 0, ok = 0;
8431     for (int i = 29; i >= 0; --i) {
8432         int x = (1 << i) ^ a;
8433         int y = (1 << i) ^ b;
8434         int z = (1 << i) ^ c;
8435         if (x < y && y < z) {
8436             a = x;
8437             b = y;
8438             c = z;
8439             ans ^= (1 << i);
8440             break;
8441         } else if (x < min(y, z) || max(x, y) < z) {
8442             a = x;
8443             b = y;
8444             c = z;
8445             ans ^= (1 << i);
8446         }
8447     }
8448     if (a < b && b < c) cout << ans;
8449     else cout << -1;
8450 }
8451
8452 int main() {
8453     ios::sync_with_stdio(false);
8454     cin.tie(0);
8455
8456     int tt = 1;
8457     cin >> tt;
8458     while (tt--) {
8459         solve();
8460         cout << '\n';
8461     }
8462     return 0;
8463 }
8464
8465 //MEXSEG
8466 // ॐ
8467 #include <bits/stdc++.h>
8468 using namespace std;
8469 #define PI 3.14159265358979323846

```

```

8470 #define ll long long int
8471
8472 const int N=1e6+5;
8473 int pos[N];
8474 int l_m[N],r_m[N];
8475
8476 inline ll f(ll m,ll len,ll n){
8477
8478     if(m>n || len<=0)
8479         return 0;
8480
8481     if(m==0)
8482         return (1LL*len*(2*n-len+1))/2;
8483
8484     int r=n-r_m[m];
8485     int l=l_m[m]-1;
8486     int sz=r_m[m]-l_m[m]+1;
8487
8488     if(sz>len){
8489         return 0;
8490     }
8491
8492     int left=len-sz;
8493     l=min(l,left);
8494     r=min(r,left);
8495
8496     int z=min(l,r)+1;
8497     ll ret=1LL*z*(z+1);
8498     ret/=2;
8499
8500     ret+=max(0LL,1LL*z*(min(max(l,r),left)-z+1));
8501     z=max(l,r);
8502     int num=min(l+r,left)-z;
8503     z=min(l,r);
8504     ret+=(1LL*(num)*(z-num+1+z))/2;
8505
8506     return ret;
8507 }
8508
8509 int main(){
8510
8511     ios_base::sync_with_stdio(false);
8512     cin.tie(0);
8513     cout.tie(0);
8514
8515
8516     int test = 1;
8517     cin>>test;
8518
8519
8520     while(test--){
8521
8522
8523         int n,q;
8524         cin>>n>>q;
8525         int p[n];
8526
8527         for(int i=0;i<n;i++){
8528             cin>>p[i];
8529             pos[p[i]]=i+1;
8530         }
8531
8532         l_m[0]=1e9;
8533         r_m[0]=-1;
8534
8535         for(int i=1;i<=n;i++){
8536             r_m[i]=max(r_m[i-1],pos[i-1]);
8537             l_m[i]=min(l_m[i-1],pos[i-1]);
8538         }

```

```

8539
8540         while(q--){
8541             int l1,l2,m1,m2;
8542             cin>>l1>>l2>>m1>>m2;
8543             cout<<f(m1,l2,n)-f(m1,l1-1,n)-(f(m2+1,l2,n)-f(m2+1,l1-1,n))<<
                '\n';
8544         }
8545
8546         // cout<<'\n';
8547
8548     }
8549     return 0;
8550 }
8551
8552 //SLAYS
8553 #include "bits/stdc++.h"
8554 using namespace std;
8555
8556 typedef long long        lol;
8557 typedef std::pair<int,int> pii;
8558 #define pb              push_back
8559 #define ub              upper_bound
8560 #define lb              lower_bound
8561 #define fo(i,l,r,d)     for (auto i=(l); (d)<0?i>(r):((d)>0?i<(r):0); i+=(d))
8562 #define all(x)           x.begin(), x.end()
8563 #define ff              first
8564 #define ss              second
8565
8566 std::mt19937 rng (std::chrono::high_resolution_clock::now().time_since_epoch().count());
8567 template <typename A, typename B> std::ostream& operator<< (std::ostream &cout, const
std::pair<A, B> &p) { return cout << p.first << ' ' << p.second; } template <typename A,
size_t n> std::ostream& operator<< (std::ostream &cout, const std::array<A, n> &v) {
for (int i = 0; i < n - 1; ++i) cout << v[i] << ' '; return (n ? cout << v.back(): cout
<< '\n'); } template <typename A> std::ostream& operator<< (std::ostream &cout, const
std::vector<A> &v) { for (int i = 0; i < v.size() - 1; ++i) cout << v[i] << ' '; return
(v.size() ? cout << v.back(): cout << '\n'); }
8568 template <typename A, typename B> std::istream& operator>> (std::istream &cin, std::pair
<A, B> &p) { cin >> p.first; return cin >> p.second; } template <typename A, size_t n>
std::istream& operator>> (std::istream &cin, std::array<A, n> &v) { assert(n); for (int
i = 0; i < n - 1; i++) cin >> v[i]; return cin >> v.back(); } template <typename A> std
::istream& operator>> (std::istream &cin, std::vector<A> &v) { assert(v.size()); for (
int i = 0; i < v.size() - 1; i++) cin >> v[i]; return cin >> v.back(); }
8569 template <typename A, typename B> auto amax (A &a, const B b){ if (b > a) a = b ; return
a; }
8570 template <typename A, typename B> auto amin (A &a, const B b){ if (b < a) a = b ; return
a; }
8571
8572 template <
8573     class Node,
8574     class Calc,
8575     bool kNearestPowOf2 = false
8576 >
8577 class Segtree {
8578 public:
8579
8580     explicit Segtree (const int n, const Node id, const Calc& F)
8581     : sz(n), N(kNearestPowOf2 ? 1 << 32 - __builtin_clz(std::max(1, sz - 1)) : sz), a(N
<< 1, id), id(id), F(F)
8582     {
8583
8584     }
8585
8586     explicit Segtree (const std::vector<Node>& x, const Node id, const Calc& F)
8587     : sz(x.size()), N(kNearestPowOf2 ? 1 << 32 - __builtin_clz(std::max(1, sz - 1)) : sz
), id(id), F(F)
8588     {
8589         a.resize(N << 1, id);
8590         std::copy(x.begin(), x.end(), a.begin() + N);
8591

```

```

8592     for (int i = N; --i; )
8593         a[i] = F(a[i << 1], a[i << 1 | 1]);
8594 }
8595
8596 void set (int i, const Node x) {
8597     // assert(0 <= i and i < sz);
8598     for (a[i += N] = x; i >>= 1; )
8599         a[i] = F(a[i << 1], a[i << 1 | 1]);
8600 }
8601
8602 Node qu (int l, int r) const {
8603     // assert(0 <= l and l <= r and r <= sz);
8604
8605     Node x = id, y = id;
8606     for (l += N, r += N; l < r; l >>= 1, r >>= 1) {
8607         if (l & 1) x = F(x, a[l++]);
8608         if (r & 1) y = F(a[--r], y);
8609     }
8610
8611     return F(x, y);
8612 }
8613
8614 // First j in [l, N] such that pred(F[l, j]) is FALSE, if pred is monotonic
8615 template<class Predicate>
8616 int max_right (int l, const Predicate& pred) const {
8617     // assert(0 <= l and l <= N and pred(id));
8618     if (l == N) return l;
8619
8620     Node prev = id, t = id;
8621     l += N;
8622
8623     do {
8624         l >>= __builtin_ctz(l);
8625         if (!pred(F(prev, a[l]))) {
8626             while (l < N)
8627                 if (pred(t = F(prev, a[l <= 1])))
8628                     prev = t, l++;
8629             return l - N;
8630         }
8631
8632         prev = F(prev, a[l++]);
8633     } while ((l & -l) != l);
8634
8635     return N;
8636 }
8637
8638 // First j in [0, r] such that pred(F[j, r]) is TRUE, if pred is monotonic
8639 template<class Predicate>
8640 int min_left (int r, const Predicate& pred) const {
8641     // assert(r > -1 and r <= N and pred(id));
8642     if (r == 0) return r;
8643
8644     Node last = id, t = id;
8645     r += N;
8646
8647     do {
8648         r--, r >>= __builtin_ctz(~r);
8649         if (r == 0) r = 1;
8650         if (!pred(F(a[r], last))) {
8651             while (r < N)
8652                 if (pred(t = F(a[(r <= 1) += 1], last)))
8653                     last = t, r--;
8654             return r + 1 - N;
8655         }
8656
8657         last = F(a[r], last);
8658     } while((r & -r) != r);
8659
8660     return 0;

```



```

8661     }
8662
8663 private:
8664
8665     const int sz;
8666     const int N;
8667     std::vector<Node> a;
8668     const Node id;
8669     const Calc F;
8670 };
8671
8672 void darling (const int kase) {
8673
8674     int n, q; cin >> n >> q;
8675     vector a(n, 0); cin >> a;
8676
8677     a.pb(-1);
8678
8679     stack<pair<int, int>> s;
8680     vector geq(n, -1);
8681
8682     s.push(pair(0, a[0]));
8683
8684     for (int i = 1; i < n; i++) {
8685         while (s.size() and s.top().ss < a[i])
8686             s.pop();
8687         if (s.size())
8688             geq[i] = s.top().ff;
8689         s.push(pair(i, a[i]));
8690     }
8691
8692     vector dp(n, n);
8693
8694     fo(i, n-1, -1, -1) {
8695         int j = geq[i];
8696         if (a[j] == a[i])
8697             dp[j] = i;
8698         else if (a[j] == a[i] + 1)
8699             amin(dp[j], dp[i]);
8700     }
8701
8702     // cout << a << '\n' << dp << '\n';
8703
8704     vector it(n, 0);
8705     iota(all(it), 0);
8706
8707     Segtree mx(it, n, [&](int x, int y){
8708         if (a[x] > a[y])
8709             return x;
8710         else if (a[x] < a[y])
8711             return y;
8712         else
8713             return min(x, y);
8714     });
8715
8716
8717     while (q--) {
8718         int l, r; cin >> l >> r, l--;
8719
8720         auto j = mx.qu(l, r);
8721
8722         if (dp[j] < r)
8723             cout << a[j] + 1 << '\n';
8724         else
8725             cout << a[j] << '\n';
8726     }
8727
8728 }
8729

```

```

8730 int main () {
8731     ios_base::sync_with_stdio(0), cin.tie(0);
8732
8733     int t; cin >> t, assert(t >= 0);
8734     for (int i = 0; t--;)
8735         darling(++i);
8736
8737 }
8738
8739 //GUESSPFMX
8740 #include "bits/stdc++.h"
8741 using namespace std;
8742
8743 typedef long long        lol;
8744 typedef std::pair<int,int> pii;
8745 #define pb               push_back
8746 #define ub               upper_bound
8747 #define lb               lower_bound
8748 #define fo(i,l,r,d)      for (auto i=(l); (d)<0?i>(r):((d)>0?i<(r):0); i+=(d))
8749 #define all(x)            x.begin(), x.end()
8750 #define ff               first
8751 #define ss               second
8752
8753 std::mt19937 rng (std::chrono::high_resolution_clock::now().time_since_epoch().count());
8754 template <typename A, typename B> std::ostream& operator<< (std::ostream &cout, const
std::pair<A, B> &p) { return cout << p.first << ' ' << p.second; } template <typename A,
size_t n> std::ostream& operator<< (std::ostream &cout, const std::array<A, n> &v) {
for (int i = 0; i < n - 1; ++i) cout << v[i] << ' '; return (n ? cout << v.back(): cout
<< '\n'); } template <typename A> std::ostream& operator<< (std::ostream &cout, const
std::vector<A> &v) { for (int i = 0; i < v.size() - 1; ++i) cout << v[i] << ' '; return
(v.size() ? cout << v.back(): cout << '\n'); }
8755 template <typename A, typename B> std::istream& operator>> (std::istream &cin, std::pair
<A, B> &p) { cin >> p.first; return cin >> p.second; } template <typename A, size_t n>
std::istream& operator>> (std::istream &cin, std::array<A, n> &v) { assert(n); for (int
i = 0; i < n - 1; i++) cin >> v[i]; return cin >> v.back(); } template <typename A> std
::istream& operator>> (std::istream &cin, std::vector<A> &v) { assert(v.size()); for (
int i = 0; i < v.size() - 1; i++) cin >> v[i]; return cin >> v.back(); }
8756 template <typename A, typename B> auto amax (A &a, const B b){ if (b > a) a = b ; return
a; }
8757 template <typename A, typename B> auto amin (A &a, const B b){ if (b < a) a = b ; return
a; }
8758
8759
8760
8761 void darling (const int kase) {
8762
8763     int n; cin >> n;
8764     vector q(n, 0);
8765     iota(all(q), 1);
8766
8767     fo(j, n, 1, -1) {
8768         cout << "? " << q << endl;
8769         int k; cin >> k;
8770
8771         if (k == -1)
8772             exit(0);
8773
8774         vector v(k, 0);
8775
8776         cin >> v;
8777
8778         fo(c, 0, n-j, 1)
8779             v.pop_back();
8780
8781         swap(q[v.back() - 1], q[j - 1]);
8782     }
8783
8784     vector p(n, 0);
8785     fo(i, 0, n, 1)

```

```

8786         p[q[i] - 1] = i + 1;
8787
8788     cout << "! " << p << endl;
8789     int verdict;
8790     cin >> verdict;
8791     assert(verdict == 1);
8792 }
8793
8794 int main () {
8795     ios_base::sync_with_stdio(0), cin.tie(0);
8796
8797     int t; cin >> t, assert(t >= 0);
8798     for (int i = 0; t--; )
8799         darling(++i);
8800
8801 }
8802
8803 //FAULTY_TREE
8804 #include "bits/stdc++.h"
8805 #include <ext/pb_ds/assoc_container.hpp>
8806 #include <ext/pb_ds/tree_policy.hpp>
8807
8808 using namespace std;
8809 using namespace __gnu_pbds;
8810
8811 #define all(x)          x.begin(), x.end()
8812 #define pb              push_back
8813 #define sz(x)           (int)(x.size())
8814 #define ll              long long
8815 #define fi              first
8816 #define se              second
8817 #define lbd             lower_bound
8818 #define ubd             upper_bound
8819
8820
8821 template <typename T>
8822 using ordered_set = tree<T, null_type,
8823     less<T>, rb_tree_tag,
8824     tree_order_statistics_node_update>;
8825
8826 const int MOD = 1e9 + 7;
8827 const double eps = 1e-10;
8828 const long long INF = 1e12;
8829 const int N = 2e5 + 10;
8830
8831 void solve() {
8832     int n;
8833     cin >> n;
8834     vector<pair<ll, int>> v(n);
8835     for (int i = 0; i < n; i++) {
8836         cin >> v[i].fi;
8837         v[i].se = i;
8838     }
8839
8840     sort(all(v));
8841
8842     ll pre = 0;
8843     for (int i = 0; i + 1 < n; i++) {
8844         if (pre > v[i + 1].fi) {
8845             cout << "YES\n";
8846             vector<ll> ans(n);
8847             for (int j = 0; j < n; j++) {
8848                 ans[v[j].se] = v[j].fi;
8849             }
8850             for (int j = 0; j < n; j++) {
8851                 cout << ans[j] << ' ';
8852             }
8853             return;
8854         }
8855     }

```

```

8855     pre += v[i].fi;
8856 }
8857
8858 if (n <= 3) {
8859     cout << "NO";
8860     return;
8861 }
8862
8863 cout << "YES\n";
8864 if (v[0].fi + v[1].fi > v[2].fi) {
8865     v[3].fi = v[2].fi;
8866     vector<ll> ans(n);
8867     for (int j = 0; j < n; j++) {
8868         ans[v[j].se] = v[j].fi;
8869     }
8870     for (int j = 0; j < n; j++) {
8871         cout << ans[j] << ' ';
8872     }
8873     return;
8874 }
8875 if (v[1].fi + v[2].fi > v[3].fi) {
8876     v[0].fi = v[2].fi;
8877     vector<ll> ans(n);
8878     for (int j = 0; j < n; j++) {
8879         ans[v[j].se] = v[j].fi;
8880     }
8881     for (int j = 0; j < n; j++) {
8882         cout << ans[j] << ' ';
8883     }
8884     return;
8885 }
8886
8887 pre = 0;
8888 int mx = 0;
8889 for (int i = 0; i + 1 < n; i++) {
8890     if (pre > v[i].fi) {
8891         v[i + 1].fi = v[i].fi;
8892         vector<ll> ans(n);
8893         for (int j = 0; j < n; j++) {
8894             ans[v[j].se] = v[j].fi;
8895         }
8896         for (int j = 0; j < n; j++) {
8897             cout << ans[j] << ' ';
8898         }
8899         return;
8900     }
8901
8902     if (pre + v[mx + 1].fi - v[mx].fi > v[i + 1].fi) {
8903         v[mx].fi = v[mx + 1].fi;
8904         vector<ll> ans(n);
8905         for (int j = 0; j < n; j++) {
8906             ans[v[j].se] = v[j].fi;
8907         }
8908         for (int j = 0; j < n; j++) {
8909             cout << ans[j] << ' ';
8910         }
8911         return;
8912     }
8913     if (v[i + 1].fi - v[i].fi > v[mx + 1].fi - v[mx].fi) {
8914         mx = i;
8915     }
8916     pre += v[i].fi;
8917 }
8918
8919 v[n - 2].fi = v[n - 1].fi;
8920 v[n - 3].fi = v[n - 1].fi;
8921 vector<ll> ans(n);
8922 for (int j = 0; j < n; j++) {
8923     ans[v[j].se] = v[j].fi;

```

```

8924     }
8925     for (int j = 0; j < n; j++) {
8926         cout << ans[j] << ' ';
8927     }
8928 }
8929
8930 int main() {
8931     ios::sync_with_stdio(false);
8932     cin.tie(0);
8933
8934     int tt = 1;
8935     cin >> tt;
8936     while (tt--) {
8937         solve();
8938         cout << '\n';
8939     }
8940     return 0;
8941 }
8942
8943 //BEAUTY_SUM
8944                                     // 3
8945 #include <bits/stdc++.h>
8946 using namespace std;
8947 #define PI 3.14159265358979323846
8948 #define ll long long int
8949
8950
8951 const int MOD = 1e9+7; // check mod
8952 struct mod_int {
8953     int val;
8954
8955     mod_int(long long v = 0) {
8956         if (v < 0)
8957             v = v % MOD + MOD;
8958
8959         if (v >= MOD)
8960             v %= MOD;
8961
8962         val = v;
8963     }
8964
8965     static int mod_inv(int a, int m = MOD) {
8966         int g = m, r = a, x = 0, y = 1;
8967
8968         while (r != 0) {
8969             int q = g / r;
8970             g %= r; swap(g, r);
8971             x -= q * y; swap(x, y);
8972         }
8973
8974         return x < 0 ? x + m : x;
8975     }
8976
8977     explicit operator int() const {
8978         return val;
8979     }
8980
8981     mod_int& operator+=(const mod_int &other) {
8982         val += other.val;
8983         if (val >= MOD) val -= MOD;
8984         return *this;
8985     }
8986
8987     mod_int& operator-=(const mod_int &other) {
8988         val -= other.val;
8989         if (val < 0) val += MOD;
8990         return *this;
8991     }
8992

```

```

8993     static unsigned fast_mod(uint64_t x, unsigned m = MOD) {
8994         #if !defined(_WIN32) || defined(_WIN64)
8995             return x % m;
8996         #endif
8997         unsigned x_high = x >> 32, x_low = (unsigned) x;
8998         unsigned quot, rem;
8999         asm("divl %4\n"
9000             : "=a" (quot), "=d" (rem)
9001             : "d" (x_high), "a" (x_low), "r" (m));
9002         return rem;
9003     }
9004
9005     mod_int& operator*=(const mod_int &other) {
9006         val = fast_mod((uint64_t) val * other.val);
9007         return *this;
9008     }
9009
9010     mod_int& operator/=(const mod_int &other) {
9011         return *this *= other.inv();
9012     }
9013
9014     friend mod_int operator+(const mod_int &a, const mod_int &b) { return mod_int(a) +=
b; }
9015     friend mod_int operator-(const mod_int &a, const mod_int &b) { return mod_int(a) -=
b; }
9016     friend mod_int operator*(const mod_int &a, const mod_int &b) { return mod_int(a) *=
b; }
9017     friend mod_int operator/(const mod_int &a, const mod_int &b) { return mod_int(a) /=
b; }
9018
9019     mod_int& operator++() {
9020         val = val == MOD - 1 ? 0 : val + 1;
9021         return *this;
9022     }
9023
9024     mod_int& operator--() {
9025         val = val == 0 ? MOD - 1 : val - 1;
9026         return *this;
9027     }
9028
9029     mod_int operator++(int32_t) { mod_int before = *this; ++*this; return before; }
9030     mod_int operator--(int32_t) { mod_int before = *this; --*this; return before; }
9031
9032     mod_int operator-() const {
9033         return val == 0 ? 0 : MOD - val;
9034     }
9035
9036     bool operator==(const mod_int &other) const { return val == other.val; }
9037     bool operator!=(const mod_int &other) const { return val != other.val; }
9038
9039     mod_int inv() const {
9040         return mod_inv(val);
9041     }
9042
9043     mod_int pow(long long p) const {
9044         assert(p >= 0);
9045         mod_int a = *this, result = 1;
9046
9047         while (p > 0) {
9048             if (p & 1)
9049                 result *= a;
9050
9051             a *= a;
9052             p >>= 1;
9053         }
9054
9055         return result;
9056     }
9057

```

```

9058     friend ostream& operator<<(ostream &stream, const mod_int &m) {
9059         return stream << m.val;
9060     }
9061     friend istream& operator >> (istream &stream, mod_int &m) {
9062         return stream>>m.val;
9063     }
9064 };
9065
9066 #include <ext/pb_ds/assoc_container.hpp>
9067 using namespace __gnu_pbds;
9068 struct custom_hash {
9069     static uint64_t splitmix64(uint64_t x) {
9070         // http://xorshift.di.unimi.it/splitmix64.c
9071         x += 0x9e3779b97f4a7c15;
9072         x = (x ^ (x >> 30)) * 0xbf58476d1ce4e5b9;
9073         x = (x ^ (x >> 27)) * 0x94d049bb133111eb;
9074         return x ^ (x >> 31);
9075     }
9076
9077     size_t operator()(uint64_t x) const {
9078         static const uint64_t FIXED_RANDOM = chrono::steady_clock::now().
            time_since_epoch().count();
9079         return splitmix64(x + FIXED_RANDOM);
9080     }
9081 };
9082
9083 gp_hash_table<int, int, custom_hash> mp;
9084
9085 const int N = 2e5+5;
9086 vector<int> adj[N];
9087 int subtr[N], a[N];
9088 mod_int ans=0;
9089 vector<pair<int, int>> vec;
9090 bool vis[N];
9091
9092 struct FenwickTree {
9093     vector<mod_int> bit; // binary indexed tree
9094     int n;
9095
9096     FenwickTree(int n) {
9097         this->n = n;
9098         bit=vector<mod_int>(n, 0);
9099     }
9100
9101     mod_int sum(int r) {
9102         mod_int ret = 0;
9103         for (; r >= 0; r = (r & (r + 1)) - 1)
9104             ret += bit[r];
9105         return ret;
9106     }
9107
9108     mod_int sum(int l, int r) {
9109         return sum(r) - sum(l - 1);
9110     }
9111
9112     void add(int idx, int delta) {
9113         for (; idx < n; idx = idx | (idx + 1))
9114             bit[idx] += delta;
9115     }
9116 };
9117
9118
9119 int getsz_cd(int v, int p) {
9120     subtr[v] = 1;
9121     for (int u : adj[v]) {
9122         if (vis[u] || u == p) continue;
9123         subtr[v] += getsz_cd(u, v);
9124     }
9125     return subtr[v];

```

```

9126     }
9127
9128     int findct_cd(int v, int p, int n) {
9129         for (int u : adj[v]) {
9130             if (!vis[u] && u!= p && subtr[u] * 2 > n) return findct_cd(u, v, n);
9131         }
9132         return v;
9133     }
9134
9135     void dfs(int v,int p,int mx,int mi){
9136
9137         vec.push_back({mx,mi});
9138         for(auto u : adj[v]){
9139             if(u==p || vis[u]){
9140                 continue;
9141             }
9142             dfs(u,v,max(a[u],mx),min(mi,a[u]));
9143         }
9144     }
9145
9146     inline mod_int solve(vector<pair<int,int>> &v){
9147
9148         mod_int ret=0;
9149         sort(v.rbegin(),v.rend());
9150
9151         vector<int> ord;
9152
9153         for(auto [mx,mi] : v){
9154             ord.push_back(mi);
9155         }
9156
9157         sort(ord.begin(),ord.end());
9158         mp.clear();
9159
9160         int curr=0;
9161
9162         for(int i=0;i<(int)ord.size();i++){
9163
9164             int temp=i;
9165             while(temp+1<(int)ord.size() && ord[temp+1]==ord[i])
9166                 temp++;
9167
9168             mp[ord[i]]=(curr++);
9169             i=temp;
9170         }
9171
9172         FenwickTree ft_mi_sum(curr+1),ft_mi_cnt(curr+1);
9173
9174
9175         for(auto [mx,mi] : v){
9176             ft_mi_cnt.add(mp[mi],1);
9177             ft_mi_sum.add(mp[mi],mi);
9178         }
9179
9180         for(auto [mx,mi] : v){
9181             ft_mi_cnt.add(mp[mi],-1);
9182             ft_mi_sum.add(mp[mi],-mi);
9183             mod_int temp=ft_mi_cnt.sum(mp[mi]+1,curr);
9184             temp*=mi;
9185             temp+=ft_mi_sum.sum(mp[mi]);
9186             temp*=mx;
9187             ret+=temp;
9188         }
9189
9190         return ret;
9191     }
9192
9193
9194     void decompose_cd(int u, int p) {

```



```

9195     int n = getsz_cd(u, p);
9196     int ct = findct_cd(u, p, n);
9197     vector<pair<int,int>> tot;
9198     tot.push_back({a[ct],a[ct]});
9199     vis[ct]=1;
9200
9201
9202     for(auto chi : adj[ct]){
9203         if(vis[chi])
9204             continue;
9205         vec.clear();
9206         dfs(chi,chi,max(a[ct],a[chi]),min(a[ct],a[chi]));
9207         ans+=solve(vec);
9208         for(auto u : vec){
9209             tot.push_back(u);
9210         }
9211     }
9212
9213     ans+=solve(tot);
9214
9215     for(auto chi : adj[ct]){
9216         if(!vis[chi])
9217             decompose_cd(chi,ct);
9218     }
9219 }
9220
9221
9222 int main(){
9223
9224     ios_base::sync_with_stdio(false);
9225     cin.tie(0);
9226     cout.tie(0);
9227
9228     int test = 1;
9229     cin>>test;
9230
9231     assert(test<=1000);
9232     int sum_n=0;
9233
9234
9235     while(test--){
9236
9237         int n;
9238         cin>>n;
9239
9240         for(int i=0;i<n;i++){
9241             cin>>a[i];
9242             adj[i].clear();
9243             vis[i]=0;
9244         }
9245
9246         for(int i=1;i<n;i++){
9247             int a,b;
9248             cin>>a>>b;
9249             --a,--b;
9250             adj[a].push_back(b);
9251             adj[b].push_back(a);
9252         }
9253
9254         ans=0;
9255         decompose_cd(0,-1);
9256         cout<<ans;
9257
9258         cout<<"\n";
9259
9260     }
9261
9262     return 0;
9263

```

```

9264 }
9265
9266 //FIND_X
9267 #include <bits/stdc++.h>
9268 using namespace std;
9269
9270 #define ll long long int
9271
9272 int main() {
9273     // your code goes here
9274
9275     int t; cin>>t;
9276     assert(t<=1e5);
9277     while(t--)
9278     {
9279         ll A, B, C, D; cin>>A>>B>>C>>D;
9280
9281         assert(A>0 && A<=1e9);
9282         assert(B>0 && B<=1e9);
9283         assert(C>0 && C<=1e9);
9284         assert(D>0 && D<=1e9);
9285         assert(A%B==C%D);
9286
9287         A%=B, C%=D;
9288
9289
9290
9291         if((A+1)%B==(C+1)%D) cout<<"1\n";
9292         else
9293         {
9294             ll ans=B*D/__gcd(B, D)-A;
9295             cout<<ans<<"\n";
9296         }
9297     }
9298
9299
9300     return 0;
9301 }
9302
9303 //MATDIF
9304 #include <iostream>
9305
9306 using namespace std;
9307
9308
9309 void ANS(int n)
9310 {
9311     int curr=2;
9312     for(int i=0; i<n; i++)
9313     {
9314         for(int j=0; j<n; j++)
9315         {
9316             if(curr>n*n) curr=1;
9317             cout<<curr<<" ";
9318             curr+=2;
9319         }
9320         cout<<endl;
9321     }
9322 }
9323
9324 int main() {
9325     // your code goes here
9326
9327     int t; cin>>t;
9328     while(t--)
9329     {
9330         int n; cin>>n;
9331         ANS(n);
9332     }

```

```

9333
9334     return 0;
9335 }
9336
9337 //TREE_GAME
9338 #include<bits/stdc++.h>
9339 using namespace std;
9340 using ll=long long;
9341 const ll inf=1e16;
9342
9343 #ifdef ANI
9344 #include "D:/DUSTBIN/local_inc.h"
9345 #else
9346 #define dbg(...) 0
9347 #endif
9348
9349 void solve(int &tot) {
9350     ll n,x,y;
9351     cin>>n>>x>>y;
9352     assert(n<=5000 && x<=n && y<=n);
9353     x--;y--;tot+=n;
9354     vector<ll> a(n);
9355     ll nax=1e9;
9356     for(ll i=0;i<n;i++) {
9357         cin>>a[i];
9358         assert(a[i]<=nax && a[i]>=1);
9359     }
9360     vector<vector<ll>> e(n);
9361     for(ll i=0;i<n-1;i++) {
9362         ll u,v;
9363         cin>>u>>v;
9364         e[u-1].push_back(v-1);
9365         e[v-1].push_back(u-1);
9366         assert(u<=n && v<=n && u>=1 && v>=1 && u!=v);
9367     }
9368
9369     auto dfs=[&](ll u,ll v,ll su,ll sv,ll pu,ll pv,ll score,auto &&dfs)->ll{ // comsute
game states
9370         su+=a[u],sv+=a[v]; score+=su>sv;
9371         if((e[u].size()==1&&u!=x) || (e[v].size()==1&&v!=y))
9372             return score;
9373         ll res=0;
9374         for(ll p:e[u]) {
9375             if(p==pu) continue;
9376             ll cur=inf;
9377             for(ll q:e[v]) {
9378                 if(q==pv)
9379                     cur=min(cur,dfs(p,q,su,sv,u,v,score,dfs));
9380             }
9381             res=max(res,cur);
9382         }
9383         return res;
9384     };
9385     cout<<dfs(x,y,0,0,-1,-1,0,dfs)<<"\n";
9386 }
9387
9388 int main() {
9389     ios_base::sync_with_stdio(false);cin.tie(NULL);cout.tie(NULL);
9390     int t;
9391     cin>>t;
9392     assert(t<=1000);
9393     int tot=0;
9394     while(t--) {
9395         solve(tot);
9396     }
9397     assert(tot<=5000);
9398 }
9399
9400 //ALIKE_THEM

```

```

9401 #include <bits/stdc++.h>
9402 using namespace std;
9403
9404 const int MOD=1e9+7;
9405
9406 #define ll long long int
9407
9408
9409
9410
9411
9412
9413 int solve(int n, int m, vector<int>p, vector<int>a)
9414 {
9415     bool fetch[n+1]{};
9416     int curr=0, c=0, x=0;
9417     for(int i=1; i<=n; i++) if(p[i]>=i) fetch[p[i]]=true, x++;
9418
9419     bool zero=false;
9420     for(int i=1; i<=n; i++)
9421     {
9422         if(fetch[i] && a[i]==0) zero=true;
9423         if(fetch[i] && a[i])
9424         {
9425             if(curr && (curr^a[i])) return 0;
9426             curr=a[i];
9427         }
9428         if(!fetch[i] && a[i]) c++;
9429     }
9430
9431     int exp=n-x+1-c;
9432     if(curr) exp--;
9433
9434
9435     ll ans=1;
9436     while(exp--) ans=(ans*m)%MOD;
9437
9438     if(curr>m && zero)
9439     {
9440         return 0;
9441     }
9442     return ans;
9443 }
9444
9445 int main()
9446 {
9447
9448     int t; cin>>t;
9449
9450     assert(t<=1e5);
9451     int total_n=0;
9452
9453     while(t--)
9454     {
9455         int n, m; cin>>n>>m;
9456         total_n+=n;
9457         assert(n>=1 && n<=2e5);
9458         assert(m>=1 && m<=1e9);
9459
9460         vector<int>p(n+1), a(n+1);
9461
9462         for(int i=1; i<=n; i++) cin>>p[i];
9463         for(int i=1; i<=n; i++) cin>>a[i];
9464
9465         bool visi[n+1]{};
9466         for(int i=1; i<=n; i++)
9467         {
9468             visi[p[i]]=true;
9469         }

```

```

9470
9471     for(int i=1; i<=n; i++)
9472     {
9473         assert(a[i]>=0 && a[i]<=1e9);
9474         assert(p[i]);
9475     }
9476
9477     cout<<solve(n, m, p, a)<<endl;
9478 }
9479 assert(total_n<=2e5);
9480
9481
9482 }
9483
9484 //GRIDMEET
9485 #include<bits/stdc++.h>
9486 using namespace std;
9487 using ll=long long;
9488 const ll inf=1e16;
9489
9490 #ifdef ANI
9491 #include "D:/DUSTBIN/local_inc.h"
9492 #else
9493 #define dbg(...) 0
9494 #endif
9495
9496 void solve(ll &tot) {
9497     ll n,k;
9498     cin>>n>>k;
9499     ll nax=1e9;
9500     assert(n<=1000 && k<=n);
9501     tot+=n;
9502     map<ll,vector<ll>> xp,yp;
9503     vector<vector<ll>> a(n,vector<ll>(2));
9504     for(int i=0;i<n;i++) {
9505         ll x,y;
9506         cin>>x>>y;
9507         xp[x].push_back(i);
9508         yp[y].push_back(i);
9509         a[i]={x,y};
9510         assert(abs(x)<=nax && abs(y)<=nax);
9511     }
9512     vector<ll> xc;
9513     for(auto el:xp) xc.push_back(el.first);
9514     ll ans=inf;
9515     for(int i=0;i<xc.size();i++) {
9516         set<pair<ll,ll>> uu,dd,uh; vector<bool> dh(n,0);
9517         ll xx=xc[i],yy=yp.begin()->first;
9518         for(auto el:yp) {
9519             auto pts=el.second;
9520             ll yi=el.first;
9521             for(ll ii:pts) {
9522                 ll xi=a[ii][0];
9523                 dd.insert({abs(xx-xi)+abs(yy-yi),ii});
9524             }
9525         }
9526         ll cur=0,mv=0,dct=0,uct=0; // mv: how much we have moved down
9527         for(auto it=yp.begin();it!=yp.end();it++) {
9528             while(!uh.empty()) {
9529                 auto it=uh.end(),jt=dd.begin();
9530                 it--;
9531                 if(jt==dd.end() or jt->first-mv>it->first+mv) break;
9532                 cur-=it->first+mv;
9533                 uu.insert(*it);
9534                 uh.erase(it);
9535             }
9536             while(dct+uh.size() < k) {
9537                 auto it=dd.begin(),jt=uu.begin();
9538                 if(it==dd.end()) {

```

```

9539         cur+=jt->first+mv;
9540         uh.insert(*jt);
9541         uu.erase(jt);
9542     } else if(jt==uu.end()) {
9543         cur+=it->first-mv;
9544         dh[it->second]=1; dct++;
9545         dd.erase(it);
9546     } else {
9547         ll gd=it->first-mv,gu=jt->first+mv;
9548         if(gd<gu) {
9549             cur+=it->first-mv;
9550             dh[it->second]=1; dct++;
9551             dd.erase(it);
9552         } else {
9553             cur+=jt->first+mv;
9554             uh.insert(*jt);
9555             uu.erase(jt);
9556         }
9557     }
9558 }
9559 ans=min(ans,cur);
9560 for(ll ii:it->second) {
9561     ll xi=a[ii][0],cost=abs(xx-xi)+abs(yy-it->first);
9562     uu.insert({cost-2*mv,ii});
9563     if(dh[ii]) {
9564         dh[ii]=0;
9565         dct--;
9566         cur-=cost-mv;
9567     } else dd.erase({cost,ii});
9568 }
9569 auto jt=it; jt++;
9570 ll delta=0;
9571 if(jt!=yp.end()) {
9572     delta=jt->first-it->first;
9573     mv+=delta;
9574 }
9575 cur+=(uh.size()-dct)*delta;
9576 }
9577 }
9578 cout<<ans<<"\n";
9579 }
9580
9581 int main() {
9582     ios_base::sync_with_stdio(false);cin.tie(NULL);cout.tie(NULL);
9583     int t;
9584     cin>>t;
9585     assert(t<=1000);
9586     ll tot=0;
9587     while(t--) {
9588         solve(tot);
9589     }
9590     assert(tot<=1000);
9591 }
9592
9593 //MODE
9594 #include<bits/stdc++.h>
9595 using namespace std;
9596 using ll=long long;
9597 const ll inf=1e16;
9598
9599 #ifdef ANI
9600 #include "D:/DUSTBIN/local_inc.h"
9601 #else
9602 #define dbg(...) 0
9603 #endif
9604
9605 vector<ll> solution(vector<ll> arr) {
9606     ll n=arr.size();
9607     vector<ll>freq(n+1);

```

```

9608     vector<pair<ll,ll>>>v;
9609     map<ll,ll>mp;
9610     for(ll i=0;i<n;i++){
9611         mp[arr[i]]++;
9612     }
9613     for(auto el:mp){
9614         ll x=el.first, y=el.second;
9615         freq[y]++;
9616     }
9617     vector<ll>ans(n+1,1e9);
9618     ans[n]=0;
9619     v.push_back({0,1e9});
9620     for(ll i=1;i<=n;i++){
9621         if(freq[i]>0){
9622             v.push_back({i,freq[i]});
9623             ans[n]+=(i-1)*freq[i];
9624         }
9625     }
9626     ll suff=0,cnt=0;
9627
9628     for(ll i=n;i>=2;i--){
9629         ll op=suff-cnt*i,curr=freq[i]+cnt,x=1,redu_ele=op;
9630         pair<ll,ll>p={i,0};
9631         ll idx=lower_bound(v.begin(),v.end(),p)-v.begin();
9632         idx--;
9633         for(ll j=1;j<=n/i;j++){
9634             if(j<=curr){
9635                 ans[j]=min(ans[j],op+(curr-j));
9636             }
9637             else{
9638                 if(v[idx].second>=x){
9639                     x++;
9640                 }
9641                 else{
9642                     idx--;
9643                     x=2;
9644                 }
9645                 ll value=max(i-v[idx].first-redu_ele,0ll);
9646                 redu_ele-=min(redu_ele,max(i-v[idx].first-value,0ll));
9647                 op+=(value);
9648                 ans[j]=min(ans[j],op);
9649             }
9650         }
9651         suff+=(freq[i]*i);
9652         cnt+=freq[i];
9653     }
9654     for(auto &x:ans) {
9655         if(x==1e9) x=-1;
9656     }
9657     return vector<ll>(ans.begin()+1,ans.end());
9658 }
9659
9660
9661 void solve() {
9662     ll t;
9663     cin>>t;
9664     assert(t<=100000);
9665     ll tot=0;
9666     while(t--){
9667         ll n;
9668         cin>>n;
9669         tot+=n;
9670         vector<ll> a(n);
9671         ll nax=1e9;
9672         for(ll i=0;i<n;i++){
9673             cin>>a[i];
9674             assert(a[i]<=nax);
9675         }
9676         vector<ll> res=solution(a);

```

```

9677         for (ll i=0;i<n;i++) {
9678             cout<<res[i]<<" \n"[i==n-1];
9679         }
9680     }
9681     assert (tot<=200000);
9682 }
9683
9684 int main() {
9685     solve();
9686 }
9687
9688 //GOOD_PERM
9689 #include<bits/stdc++.h>
9690 using namespace std;
9691
9692 mt19937 rng (chrono::steady_clock::now().time_since_epoch().count());
9693
9694 #define int    long long
9695 #define F      first
9696 #define S      second
9697 #define pb     push_back
9698 #define endl   "\n"
9699
9700 const int mod=1e9+7;
9701 const int mod1=998244353;
9702 const int inf=1e18;
9703 const long double pi=2*acos(0.0);
9704 const long double eps=1e-9;
9705 const int N=2e5;
9706
9707 int fact[N+1];
9708
9709 int power(int a,int b){
9710     int res=1;
9711     while(b){
9712         if(b&1){
9713             res=(res*a)%mod;
9714         }
9715         a=(a*a)%mod;
9716         b/=2;
9717     }
9718     return res;
9719 }
9720
9721 int C(int n,int r){
9722     return (fact[n]*power((fact[r]*fact[n-r])%mod,mod-2))%mod;
9723 }
9724
9725 void solve(){
9726     int n;
9727     cin>>n;
9728     map<int,int>mp1,mp2;
9729     for (int i=0;i<n;i++){
9730         int x;
9731         cin>>x;
9732         mp1[x]++;
9733     }
9734     for (int i=0;i<n;i++){
9735         int x;
9736         cin>>x;
9737         x=abs(x);
9738         mp2[x]++;
9739     }
9740     if (mp1.size()==1 and mp2.size()==1){
9741         cout<<fact[n]<<endl;
9742         return;
9743     }
9744     map<int,int>mp3=mp2;

```



```

9746     int ans=0;
9747     int curr=1;
9748     int to_achieve1=(*mp1.begin()).F+(*mp3.rbegin()).F;
9749     for(auto x:mp1){
9750         int required=abs(to_achieve1-x.F);
9751         if(mp3[required]<x.S){
9752             curr*=0;
9753             break;
9754         }
9755         else{
9756             curr=(curr*C(mp3[required],x.S))%mod;
9757             curr=(curr*fact[x.S])%mod;
9758             mp3[required]-=x.S;
9759         }
9760     }
9761     ans+=curr;
9762     mp3=mp2;
9763     int to_achieve2=(*mp1.rbegin()).F-(*mp3.rbegin()).F;
9764     curr=0;
9765     if(to_achieve1!=to_achieve2){
9766         curr=1;
9767         for(auto x:mp1){
9768             int required=abs(to_achieve2-x.F);
9769             if(mp3[required]<x.S){
9770                 curr*=0;
9771                 break;
9772             }
9773             else{
9774                 curr=(curr*C(mp3[required],x.S))%mod;
9775                 curr=(curr*fact[x.S])%mod;
9776                 mp3[required]-=x.S;
9777             }
9778         }
9779     }
9780     ans+=curr;
9781     cout<<ans%mod<<endl;
9782 }
9783 int32_t main(){
9784     ios_base::sync_with_stdio(false);
9785     cin.tie(NULL);
9786     cout.tie(NULL);
9787     fact[0]=1;
9788     for(int i=1;i<=N;i++){
9789         fact[i]=(fact[i-1]*i)%mod;
9790     }
9791     int t=1;
9792     cin>>t;
9793     while(t--){
9794         solve();
9795     }
9796     return 0;
9797 }
9798
9799 //BSTRING
9800 #include <iostream>
9801 using namespace std;
9802 #define endl '\n'
9803 #define int long long
9804
9805 const int mod = 1e9+7;
9806
9807 int binpow(int a,int b) {
9808     if(b<0) return 0;
9809     int res = 1;
9810     while(b > 0) {
9811         if(b & 1) res = res*a%mod;
9812         a=a*a%mod;
9813         b>>=1;
9814     }

```

```

9815     return res;
9816 }
9817
9818 void solve() {
9819     int n;
9820     cin>>n;
9821
9822     string s;
9823     cin>>s;
9824
9825     int pre0 = 0, pre1 = 0;
9826     int ans = s.size();
9827
9828     int inv2 = binpow(2,mod-2);
9829     int inv2Pow = inv2;
9830     int pow2 = 1;
9831
9832     for(auto u:s) {
9833         inv2Pow = inv2Pow*inv2%mod;
9834         pow2 = pow2*2%mod;
9835
9836         if(u == '1') {
9837             ans = (ans + pre1*pow2)%mod;
9838             pre1 = (pre1 + inv2Pow)%mod;
9839         }
9840         else {
9841             ans = (ans + pre0*pow2)%mod;
9842             pre0 = (pre0 + inv2Pow)%mod;
9843         }
9844     }
9845
9846     cout<<ans<<endl;
9847 }
9848
9849 signed main() {
9850     ios_base::sync_with_stdio(false);
9851     cin.tie(nullptr);cout.tie(nullptr);
9852     int testcases = 1;
9853     cin>>testcases;
9854     while(testcases--) solve();
9855     return 0;
9856 }
9857
9858 //MIND_IF
9859 #include <iostream>
9860 #include <string>
9861 #include <set>
9862 #include <map>
9863 #include <stack>
9864 #include <queue>
9865 #include <vector>
9866 #include <utility>
9867 #include <iomanip>
9868 #include <sstream>
9869 #include <bitset>
9870 #include <cstdlib>
9871 #include <iterator>
9872 #include <algorithm>
9873 #include <cstdio>
9874 #include <cctype>
9875 #include <cmath>
9876 #include <math.h>
9877 #include <ctime>
9878 #include <cstring>
9879 #include <unordered_set>
9880 #include <unordered_map>
9881 #include <cassert>
9882 #define int long long int
9883 #define pb push_back

```

```

9884 #define mp make_pair
9885 #define mod 1000000007
9886 #define vl vector<ll>
9887 #define all(c) (c).begin(),(c).end()
9888 using namespace std;
9889
9890 const int N=500023;
9891 bool vis[N];
9892 vector<int> adj[N];
9893 long long readInt(long long l,long long r,char endd){
9894     long long x=0;
9895     int cnt=0;
9896     int fi=-1;
9897     bool is_neg=false;
9898     while(true){
9899         char g=getchar();
9900         if(g=='-'){
9901             assert(fi!=-1);
9902             is_neg=true;
9903             continue;
9904         }
9905         if('0'<=g && g<='9'){
9906             x*=10;
9907             x+=g-'0';
9908             if(cnt==0){
9909                 fi=g-'0';
9910             }
9911             cnt++;
9912             assert(fi!=0 || cnt==1);
9913             assert(fi!=0 || is_neg==false);
9914
9915             assert(!(cnt>19 || ( cnt==19 && fi>1) ));
9916         } else if(g==endd){
9917             if(is_neg){
9918                 x= -x;
9919             }
9920
9921             if(!(l <= x && x <= r))
9922             {
9923                 cerr << l << ' ' << r << ' ' << x << '\n';
9924                 assert(1 == 0);
9925             }
9926
9927             return x;
9928         } else {
9929             assert(false);
9930         }
9931     }
9932 }
9933
9934 string readString(int l,int r,char endd){
9935     string ret="";
9936     int cnt=0;
9937     while(true){
9938         char g=getchar();
9939         assert(g!=-1);
9940         if(g==endd){
9941             break;
9942         }
9943         cnt++;
9944         ret+=g;
9945     }
9946     assert(l<=cnt && cnt<=r);
9947     return ret;
9948 }
9949
9950 long long readIntSp(long long l,long long r){
9951     return readInt(l,r,' ');
9952 }

```

```

9953
9954 long long readIntLn(long long l,long long r){
9955     return readInt(l,r,'\n');
9956 }
9957
9958 string readStringLn(int l,int r){
9959     return readString(l,r,'\n');
9960 }
9961
9962 string readStringSp(int l,int r){
9963     return readString(l,r,' ');
9964 }
9965
9966 int sumN = 0;
9967 int check(vector<int> &a){
9968     vector<int> b = a;
9969     sort(b.begin(), b.end());
9970     int minm = 1e18;
9971     bool allAscending = true;
9972     bool allDescending = true;
9973     for(int i = 0; i<a.size(); i++){
9974         if(a[i] != b[i]){
9975             allAscending = false;
9976         }
9977         if(a[i] != b[a.size()-1-i]){
9978             allDescending = false;
9979         }
9980         if(i>0){
9981             minm = min(minm, abs(a[i]-a[i-1]));
9982         }
9983     }
9984     if(allAscending || allDescending){
9985         return 1e18;
9986     }
9987     return minm;
9988 }
9989 void solve()
9990 {
9991     int n=readInt(2,100000,'\n');
9992     sumN += n;
9993     int a[n];
9994     unordered_set<int> s;
9995     for(int i=0;i<n-1;i++){
9996         a[i]=readInt(1,1000000000,' ');
9997         s.insert(a[i]);
9998     }
9999     a[n-1]=readInt(1,1000000000,'\n');
10000     s.insert(a[n-1]);
10001     if(s.size()==1 || n==2){
10002         cout<<-1<<'\n';
10003         return;
10004     }
10005     sort(a,a+n);
10006     vector<int> ans1, ans2;
10007
10008     for(int i = 1; i<n; i++){
10009         ans1.push_back(a[i]);
10010     }
10011     ans1.push_back(a[0]);
10012     ans2.push_back(a[n-1]);
10013     for(int i = 0; i<n-1; i++){
10014         ans2.push_back(a[i]);
10015     }
10016
10017     if(check(ans1) < check(ans2)){
10018         for(int i = 0; i<n-1; i++){
10019             cout<<ans1[i]<<" ";
10020         }
10021         cout<<ans1[n-1]<<'\n';

```

```

10022     }else{
10023         for(int i = 0; i<n-1; i++){
10024             cout<<ans2[i]<<" ";
10025         }
10026         cout<<ans2[n-1]<<'\n';
10027     }
10028     cerr << check(ans1) <<' ' <<check(ans2)<<endl;
10029 }
10030 int32_t main()
10031 {
10032     #ifndef ONLINE_JUDGE
10033     freopen("input.txt", "r", stdin);
10034     freopen("output.txt", "w", stdout);
10035     #endif
10036     ios_base::sync_with_stdio(false);
10037     cin.tie(NULL),cout.tie(NULL);
10038     int T=readInt(1,200,'\n');
10039     while(T--){
10040         solve();
10041         // cout<<'\n';
10042     }
10043     cerr << sumN << endl;
10044     assert(sumN<=200000);
10045     assert(getchar()==-1);
10046     cerr << "Time : " << 1000 * ((double)clock()) / (double)CLOCKS_PER_SEC << "ms\n";
10047     return 0;
10048 }
10049
10050 //DISTMAT
10051 //clear adj and visited vector declared globally after each test case
10052 //check for long long overflow
10053 //Mod wale question mein last mein if dalo ie. Ans<0 then ans+=mod;
10054 //Incse of close mle change language to c++17 or c++14
10055 //Check ans for n=1
10056 // #pragma GCC target ("avx2")
10057 // #pragma GCC optimize ("O3")
10058 // #pragma GCC optimize ("unroll-loops")
10059 #include <bits/stdc++.h>
10060 #include <ext/pb_ds/assoc_container.hpp>
10061 #define int long long
10062 #define IOS std::ios::sync_with_stdio(false);
10063 cin.tie(NULL);cout.tie(NULL);cout.precision(dbl::max_digits10);
10064 #define pb push_back
10065 #define mod 1000000007ll //99824435311
10066 #define lld long double
10067 #define mii map<int, int>
10068 #define pii pair<int, int>
10069 #define ll long long
10070 #define ff first
10071 #define ss second
10072 #define all(x) (x).begin(), (x).end()
10073 #define rep(i,x,y) for(int i=x; i<y; i++)
10074 #define fill(a,b) memset(a, b, sizeof(a))
10075 #define vi vector<int>
10076 #define setbits(x) __builtin_popcountll(x)
10077 #define print2d(dp,n,m) for(int i=0;i<=n;i++){for(int j=0;j<=m;j++)cout<<dp[i][j]<<"
10078 ";cout<<"\n";}
10079 typedef std::numeric_limits< double > dbl;
10080 using namespace __gnu_pbds;
10081 using namespace std;
10082 typedef tree<int, null_type, less<int>, rb_tree_tag, tree_order_statistics_node_update>
10083 indexed_set;
10084 //member functions :
10085 //1. order_of_key(k) : number of elements strictly lesser than k
10086 //2. find_by_order(k) : k-th element in the set
10087 const long long N=200005, INF=2000000000000000000;
10088 const int inf=2e9 + 5;
10089 lld pi=3.1415926535897932;
10090 int lcm(int a, int b)

```

```

10088 {
10089     int g=__gcd(a, b);
10090     return a/g*b;
10091 }
10092 int power(int a, int b, int p)
10093 {
10094     if(a==0)
10095         return 0;
10096     int res=1;
10097     a%=p;
10098     while(b>0)
10099     {
10100         if(b&1)
10101             res=(1ll*res*a)%p;
10102         b>>=1;
10103         a=(1ll*a*a)%p;
10104     }
10105     return res;
10106 }
10107 mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());
10108
10109 int getRand(int l, int r)
10110 {
10111     uniform_int_distribution<int> uid(l, r);
10112     return uid(rng);
10113 }
10114 void solve(int n)
10115 {
10116     if(n==2) {
10117         cout << "-1" << endl;
10118         return;
10119     }
10120     string s = "";
10121     for(int i=1;i<=n;i++) {
10122         s += "0";
10123     }
10124     cout<<s<<endl;
10125     s.clear();
10126     for(int i = n-2; i >= 0; i--){
10127         s += "1";
10128         for(int j = 0; j<n-1; j++){
10129             if(j == i) s += "1";
10130             else s += "0";
10131         }
10132         cout<<s<<endl;
10133         s.clear();
10134     }
10135 }
10136 int32_t main()
10137 {
10138     // IOS;
10139     int t;
10140     cin>>t;
10141     while(t-->0)
10142     {
10143         int n;
10144         cin>>n;
10145         if(n<13)
10146             solve(n);
10147         else
10148         {
10149             for(int i=0;i<n;i++)
10150             {
10151                 for(int j=0;j<n;j++)
10152                     cout<<getRand(0, 1);
10153                 cout<<"\n";
10154             }
10155         }
10156     }

```

```

10157     }
10158
10159     //SORTSET
10160     #include <iostream>
10161     #include <string>
10162     #include <set>
10163     #include <map>
10164     #include <stack>
10165     #include <queue>
10166     #include <vector>
10167     #include <utility>
10168     #include <iomanip>
10169     #include <sstream>
10170     #include <bitset>
10171     #include <cstdlib>
10172     #include <iterator>
10173     #include <algorithm>
10174     #include <cstdio>
10175     #include <cctype>
10176     #include <cmath>
10177     #include <math.h>
10178     #include <ctime>
10179     #include <cstring>
10180     #include <unordered_set>
10181     #include <unordered_map>
10182     #include <cassert>
10183     #define int long long int
10184     #define pb push_back
10185     #define mp make_pair
10186     #define mod 1000000007
10187     #define vl vector<ll>
10188     #define all(c) (c).begin(), (c).end()
10189     using namespace std;
10190
10191     const int N=500023;
10192     bool vis[N];
10193     vector<int> adj[N];
10194     long long readInt(long long l, long long r, char endd){
10195         long long x=0;
10196         int cnt=0;
10197         int fi=-1;
10198         bool is_neg=false;
10199         while(true){
10200             char g=getchar();
10201             if(g=='-'){
10202                 assert(fi==-1);
10203                 is_neg=true;
10204                 continue;
10205             }
10206             if('0'<=g && g<='9'){
10207                 x*=10;
10208                 x+=g-'0';
10209                 if(cnt==0){
10210                     fi=g-'0';
10211                 }
10212                 cnt++;
10213                 assert(fi!=0 || cnt==1);
10214                 assert(fi!=0 || is_neg==false);
10215
10216                 assert(!(cnt>19 || ( cnt==19 && fi>1) ));
10217             } else if(g==endd){
10218                 if(is_neg){
10219                     x= -x;
10220                 }
10221
10222                 if(!(l <= x && x <= r))
10223                 {
10224                     cerr << l << ' ' << r << ' ' << x << '\n';
10225                     assert(1 == 0);

```

```

10226         }
10227
10228         return x;
10229     } else {
10230         assert(false);
10231     }
10232 }
10233 }
10234 string readString(int l,int r,char endd){
10235     string ret="";
10236     int cnt=0;
10237     while(true){
10238         char g=getchar();
10239         assert(g!=-1);
10240         if(g==endd){
10241             break;
10242         }
10243         cnt++;
10244         ret+=g;
10245     }
10246     assert(l<=cnt && cnt<=r);
10247     return ret;
10248 }
10249 long long readIntSp(long long l,long long r){
10250     return readInt(l,r,' ');
10251 }
10252 long long readIntLn(long long l,long long r){
10253     return readInt(l,r,'\n');
10254 }
10255 string readStringLn(int l,int r){
10256     return readString(l,r,'\n');
10257 }
10258 string readStringSp(int l,int r){
10259     return readString(l,r,' ');
10260 }
10261
10262 int power(int a, int b, int m){
10263     if(a == 0)
10264         return 0;
10265     if(b == 0)
10266         return 1;
10267     int res = 1;
10268     while(b){
10269         if(b&1){
10270             res = (res*a)%m;
10271         }
10272         a = (a*a)%m;
10273         b >>= 1;
10274     }
10275     return res;
10276 }
10277
10278 int sumN = 0;
10279 void solve()
10280 {
10281     int n = readInt(1,100000,'\n');
10282     sumN += n;
10283     int a[n];
10284     for(int i=0; i<n-1; i++){
10285         a[i] = readInt(1,1000000000,' ');
10286     }
10287     a[n-1] = readInt(1,1000000000,'\n');
10288
10289     unordered_map<int, int> freq;
10290     int maxFreq = 0;
10291     for(int i=0; i<n; i++){
10292         freq[a[i]]++;
10293         maxFreq = max(maxFreq, freq[a[i]]);
10294     }

```



```

10295     int count[n+1] = {0};
10296     for(auto i: freq){
10297         count[i.second]++;
10298     }
10299
10300     int dist = freq.size();
10301     int total = 0;
10302     int ans = 0;
10303     int maxReached = 1;
10304     for(int i = 1; i<=maxFreq; i++){
10305         ans = (ans + (((maxReached*(dist-total))%mod)*power(i, dist-total-1, mod)))%mod;
10306         total += count[i];
10307         maxReached = (maxReached * power(i+1, count[i], mod))%mod;
10308     }
10309
10310     cout << ans << '\n';
10311 }
10312
10313 int32_t main()
10314 {
10315     #ifndef ONLINE_JUDGE
10316     freopen("input.txt", "r", stdin);
10317     freopen("output.txt", "w", stdout);
10318     #endif
10319     ios_base::sync_with_stdio(false);
10320     cin.tie(NULL),cout.tie(NULL);
10321     int T=readInt(1,20000,'\n');
10322     while(T--){
10323         solve();
10324         // cout<<'\n';
10325     }
10326     cerr << sumN << '\n';
10327     assert(sumN <= 200000);
10328     assert(getchar()==-1);
10329     cerr << "Time : " << 1000 * ((double)clock()) / (double)CLOCKS_PER_SEC << "ms\n";
10330 }
10331
10332 //HORDECHESS
10333
10334 #include <bits/stdc++.h>
10335 using namespace std;
10336 #define int long long
10337 #define MOD 1000000007
10338
10339 int power(int a,int b)
10340 {
10341     if(b==0)
10342         return 1;
10343     else
10344     {
10345         int x=power(a,b/2);
10346         int y=(x*x)%MOD;
10347         if(b%2)
10348             y=(y*a)%MOD;
10349         return y;
10350     }
10351 }
10352
10353 void solve(int tc)
10354 {
10355     int n;
10356     cin >> n;
10357     int pawns,arrangements;
10358     if(n%4==0)
10359         pawns = (n-1)*n/2;
10360     else if(n%4==1 || n%4==3)
10361         pawns = (n-1)*(n+1)/2;
10362     else
10363         pawns = (n-1)*(n+2)/2;

```

```

10364     if(n%4==0)
10365         arrangements = 1;
10366     else if(n%4==1)
10367         arrangements = power((n-1)/4,n-1);
10368     else if(n%4==2)
10369         arrangements = power((n+2)/4,2*(n-1));
10370     else
10371         arrangements = power((n+5)/4,n-1);
10372     cout << pawns << " " << arrangements << '\n';
10373 }
10374
10375 int32_t main()
10376 {
10377     ios_base::sync_with_stdio(false);
10378     cin.tie(NULL);
10379     cout.tie(NULL);
10380     int tc=1;
10381     cin >> tc;
10382     for(int ttc=1;ttc<=tc;ttc++)
10383         solve(ttc);
10384     return 0;
10385 }
10386
10387 //BINANDTER
10388 #include <bits/stdc++.h>
10389 using namespace std;
10390
10391 #define ll long long int
10392 #define all(vec) vec.begin(), vec.end()
10393 #define endl "\n"
10394 #define pb push_back
10395 #define yes cout << "YES" << endl;
10396 #define no cout << "NO" << endl;
10397 #define ff first
10398 #define ss second
10399 #define flush cout << flush;
10400 // #define N 1e5 + 1
10401 #define PI 3.141592653589793238462643383279
10402 #define IOS \
10403     ios_base::sync_with_stdio(0); \
10404     cin.tie(0); \
10405     cout.tie(0);
10406
10407 ll mod_mul(ll a, ll b, ll m) {a = a % m; b = b % m; return ((a * b) % m + m) % m;}
10408 ll mod_add(ll a, ll b, ll m) {a = a % m; b = b % m; return ((a + b) % m + m) % m;}
10409 long long power(ll x, ll y)
10410 {
10411     ll temp;
10412     if (y == 0)
10413         return 1;
10414     temp = power(x, y / 2);
10415     if (y % 2 == 0)
10416         return temp * temp;
10417     else
10418         return x * temp * temp;
10419 }
10420 vector<ll>pre(18);
10421 void _segfault_()
10422 {
10423     ll n;
10424     cin >> n;
10425     vector<ll>vec(18);
10426     for (ll i = 0; i <= 17; i++) {
10427         vec[i] = i;
10428     }
10429     ll m = n;
10430     ll o = 0;
10431     while (m > 0) {
10432         if (m % 2 == 1)

```

```

10433         o++;
10434         m /= 2;
10435     }
10436     ll mn = LLONG_MAX;
10437     mn = min(mn, o);
10438     for (ll i = 1; i < (1ll << 17); i++) {
10439         ll sum = 0;
10440         ll cnt = 0;
10441         vector<ll>a;
10442         ll j = i;
10443         ll one = 0;
10444         while (j > 0) {
10445             if (j % 2 == 1) one++;
10446             a.pb(j % 2);
10447             j /= 2;
10448         }
10449         for (ll j = 0; j < a.size(); j++) {
10450             if (a[j] == 1)
10451                 sum += pre[vec[j]];
10452         }
10453         if (sum > n) break;
10454         else {
10455             ll b = n - sum;
10456             ll c = b;
10457             while (b > 0) {
10458                 if (b % 2)
10459                     cnt++;
10460                 b /= 2;
10461             }
10462             cnt += one;
10463             mn = min(mn, cnt);
10464         }
10465     }
10466     if (mn == LLONG_MAX)
10467         cout << -1 << endl;
10468     else
10469         cout << mn << endl;
10470 }
10471 int32_t main(int argc, char const * argv[])
10472 {
10473     // int32_t for returning val 32 bit integer always
10474     IOS
10475     clock_t z = clock();
10476     cout.setf(ios::fixed, ios::floatfield);
10477     cout.setf(ios::showpoint);
10478     cout << setprecision(20);
10479     int t=1;
10480     cin >> t;
10481     ll i = 0;
10482     while (i < pre.size()) {
10483         pre[i] = power(3, i);
10484         i++;
10485     }
10486     while (t--)
10487     {
10488         // cout << "Case #" << a << ": ";
10489         _segfault_();
10490         // a++;
10491     }
10492     cerr << "Run Time : " << ((double)(clock() - z) / CLOCKS_PER_SEC);
10493     return 0;
10494 }
10495
10496 //ASFA
10497 #include <bits/stdc++.h>
10498 #include<string>
10499 #define int long long
10500 using namespace std;
10501

```

```

10502 signed main()
10503 {
10504     int t;
10505     cin>>t;
10506     while(t--)
10507     {
10508
10509         int n;
10510         cin>>n;
10511         vector< int > v(n);
10512         for(auto &x:v)
10513             cin>>x;
10514         int odd=0,even=0;
10515         for(auto i:v)
10516         {
10517             if(i%2)
10518                 ++odd;
10519             else
10520                 ++even;
10521         }
10522         if(n%2)
10523             cout<<-1;
10524         else {
10525             if(even==odd)
10526                 cout<<0;
10527             else if(even>odd){
10528                 if(odd)
10529                     cout<<(even-odd)/2;
10530                 else
10531                     cout<<-1;
10532             }
10533             else{
10534                 int ans=odd-even;
10535                 if(ans%4==0)
10536                     cout<<ans/4;
10537                 else if(odd==2)
10538                     cout<<-1;
10539                 else
10540                     cout<<(ans-2)/4+2;
10541             }
10542         }
10543         cout<<"\n";
10544     }
10545     return 0;
10546 }
10547
10548 //TO START START80
10549
10550 //C8KBFTREE
10551
10552 /*
10553  template by c8kbf
10554  */
10555
10556 // macOS doesn't have <bits/++.h> (shame)
10557 #include <cstdlib>
10558
10559 #include <iostream>
10560 #include <cstdio>
10561 #include <iomanip>
10562 #include <fstream>
10563
10564 #include <cmath>
10565 #include <cstring>
10566 #include <ctime>
10567
10568 #include <deque>
10569 #include <string>
10570 #include <stack>

```

```

10571 #include <vector>
10572 #include <map>
10573 #include <queue>
10574 #include <list>
10575 #include <set>
10576 #include <unordered_map>
10577 #include <unordered_set>
10578 #include <bitset>
10579
10580 #include <algorithm>
10581 #include <numeric>
10582 #include <random>
10583 #include <functional>
10584
10585 //dont worry bout me, i'm not high
10586 #define ef else if
10587 #define leave exit(0);
10588
10589 #define v(x) vector<x >
10590 #define v2(x) vector<vector<x > >
10591 #define v3(x) vector<vector<vector<x > > >
10592
10593 #define q(x) queue<x >
10594 #define dq(x) deque<x >
10595 #define s(x) set<x >
10596 #define st(x) stack<x >
10597 #define ms(x) multiset<x >
10598 #define m(x, y) map<x , y >
10599 #define b(x) bitset<x >
10600 #define l(x) list<x >
10601
10602 #define ss(x) v(_)(x+8, 0)
10603 #define ssz(type, x) v(type)(x+8, 0)
10604 #define s2(x, y) v2(_)(x+8, v(_)(y+8, 0))
10605 #define s2z(type, x, y) v2(type)(x+8, v(type)(y+8, 0))
10606 #define s3(x, y, z) v3(_)(x+8, v2(_)(y+8, v(_)(z+8, 0)))
10607 #define s3z(type, x, y, z) v3(type)(x+8, v2(type)(y+8, v(type)(z+8, 0)))
10608 #define rd(a, sz) for(_ i = 1; i <= sz; ++i) a[i] = read();
10609 #define wr(a, sz) for(_ i = 1; i <= sz; ++i) writesc(a[i]); clr();
10610
10611 #define i(x) x::iterator
10612
10613 #define pr(x, y) pair< x, y >
10614 #define mp(x, y) make_pair(x, y)
10615
10616 using namespace std;
10617
10618 //weirdest typedefs ever??
10619 typedef long long _;
10620 typedef int _0;
10621 typedef double _D;
10622 typedef unsigned long long u_;
10623 typedef string str;
10624 typedef vector<_> v_;
10625 typedef pair<_, _> _p;
10626 typedef const long long constant;
10627
10628 //fastIO cos why not
10629 inline _ read() {
10630     _ x = 0, f = 1;
10631     char ch = getchar();
10632     for(; !(ch >= '0' && ch <= '9'); ch = getchar()) if(ch == '-') f *= -1;
10633     for(; (ch >= '0' && ch <= '9'); ch = getchar()) x = (x<<3)+(x<<1)+(ch^48);
10634     return x*f;
10635 }
10636
10637 inline bool read(_ & x, v(char) tl = {'\n', EOF}) {
10638     x = 0;
10639     _ f = 1;

```

```

10640     char ch = getchar();
10641     for(; !(ch >= '0' && ch <= '9'); ch = getchar()) if(ch == '-') f *= -1;
10642     for(; (ch >= '0' && ch <= '9'); ch = getchar()) x = (x<<3)+(x<<1)+(ch^48);
10643     x *= f;
10644     if(ch == '\r') ch = getchar();
10645     return !count(tl.begin(), tl.end(), ch);
10646 }
10647
10648 inline void read(char * a, v(char) tl = {' ', '\n', '\r', '\t', '\0', EOF}, v(char) skp
= {' ', '\n', '\r', '\t'}) {
10649     char ch = getchar();
10650     for(; count(skp.begin(), skp.end(), ch); ) ch = getchar();
10651     for(; !count(tl.begin(), tl.end(), ch); ch = getchar()) {
10652         *a = ch;
10653         ++a;
10654     }
10655     *a = '\0';
10656     return;
10657 }
10658
10659 inline void read(str & a, v(char) tl = {' ', '\n', '\r', '\t', '\0', EOF}, v(char) skp =
{' ', '\n', '\r', '\t'}) {
10660     a.clear();
10661     char ch = getchar();
10662     for(; count(skp.begin(), skp.end(), ch); ) ch = getchar();
10663     for(; !count(tl.begin(), tl.end(), ch); ch = getchar()) a += ch;
10664     return;
10665 }
10666
10667 inline void read(vector<reference_wrapper<_> > a) {
10668     for(_ & i : a) i = read();
10669     return;
10670 }
10671
10672 inline void read(_p & x) {
10673     x.first = read();
10674     x.second = read();
10675     return;
10676 }
10677
10678 inline char getDg() {
10679     char ch = getchar();
10680     for(; !(ch >= '0' && ch <= '9'); ) ch = getchar();
10681     return ch;
10682 }
10683
10684 inline char getLw() {
10685     char ch = getchar();
10686     for(; !(ch >= 'a' && ch <= 'z'); ) ch = getchar();
10687     return ch;
10688 }
10689
10690 inline char getUp() {
10691     char ch = getchar();
10692     for(; !(ch >= 'A' && ch <= 'Z'); ) ch = getchar();
10693     return ch;
10694 }
10695
10696 inline char getLtr() {
10697     char ch = getchar();
10698     for(; !((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z')); ) ch = getchar();
10699     return ch;
10700 }
10701
10702 inline char gc() {
10703     char ch = getchar();
10704     for(; ch == '\n' || ch == '\r' || ch == ' '; ) ch = getchar();
10705     return ch;
10706 }

```

```

10707
10708 inline void write(_ x) {
10709     if(x < 0) {
10710         putchar('-');
10711         write(-x);
10712         return;
10713     }
10714     if(x > 9) write(x/10);
10715     putchar((x%10)^48);
10716     return;
10717 }
10718
10719 inline void write(_0 x) {
10720     write((_x);
10721     return;
10722 }
10723
10724 inline void write(char const * a) {
10725     for(_ i = 0; a[i]; ++i) putchar(a[i]);
10726     return;
10727 }
10728
10729 inline void write(const str a) {
10730     write(a.c_str());
10731     return;
10732 }
10733
10734 inline void write(char ch) {
10735     putchar(ch);
10736     return;
10737 }
10738
10739 inline void write(_p a, char const * b = " ") {
10740     write(a.first);
10741     write(b);
10742     write(a.second);
10743     return;
10744 }
10745
10746 inline void write(v_ a, char const * b = " ") {
10747     bool fs = false;
10748     for(_ i : a) {
10749         if(!fs) fs = true;
10750         else write(b);
10751         write(i);
10752     }
10753     return;
10754 }
10755
10756 inline void clr() {
10757     putchar(10);
10758     return;
10759 }
10760
10761 inline void flsh(bool nl = true) {
10762     if(nl) clr();
10763     fflush(stdout);
10764     return;
10765 }
10766
10767 inline void spc() {
10768     putchar(32);
10769     return;
10770 }
10771
10772 template <class tp>
10773 inline void writeln(tp x) {
10774     write(x);
10775     clr();

```

```

10776 }
10777
10778 inline void writeln(_p a, char const * b = " ") {
10779     write(a, b);
10780     clr();
10781     return;
10782 }
10783
10784 inline void writeln(v_ a, char const * b = " ") {
10785     write(a, b);
10786     clr();
10787     return;
10788 }
10789
10790 template <class tp>
10791 inline void writesc(tp x) {
10792     write(x);
10793     spc();
10794 }
10795
10796 inline void writesc(_p a, char const * b = " ") {
10797     write(a);
10798     spc();
10799     return;
10800 }
10801
10802 template <class tp>
10803 inline void writeflsh(tp x, bool nl = true) {
10804     write(x);
10805     flsh(nl);
10806 }
10807
10808 inline void writeflsh(_p a, char const * b = " ", bool nl = true) {
10809     write(a, b);
10810     flsh(nl);
10811     return;
10812 }
10813
10814 inline void yes(_ a = 1) {
10815     write(a & 1 ? 'Y' : 'y');
10816     write(a & 2 ? 'E' : 'e');
10817     write(a & 4 ? 'S' : 's');
10818     clr();
10819     return;
10820 }
10821
10822 inline void no(_ a = 1) {
10823     write(a & 1 ? 'N' : 'n');
10824     write(a & 2 ? 'O' : 'o');
10825     clr();
10826     return;
10827 }
10828
10829 //loop systems
10830 inline v_ rg(_ r, _ l = 1, _ d = 1) {
10831     v_ rv;
10832     for(_ i = l; i <= r; i += d) rv.push_back(i);
10833     return rv;
10834 }
10835
10836 inline v_ dg(_ r, _ l = 1, _ d = -1) {
10837     v_ rv;
10838     for(_ i = r; i >= l; i += d) rv.push_back(i);
10839     return rv;
10840 }
10841
10842 inline void AC();
10843 int main(int argc, char * argv[]) {
10844

```



```

10845 // freopen("/Users/ryanzhang/Dropbox/Problemsetting/Problems In Progress/Codechef -
10846 C8KBFTREE/data/3.in", "r", stdin);
10847 // #define file_IO
10848 #ifndef file_IO
10849     str fileN = "";
10850     freopen((fileN+".in").c_str(), "r", stdin);
10851     freopen((fileN+".out").c_str(), "w", stdout);
10852 #endif
10853
10854     #define multiple_testcases
10855 #ifndef multiple_testcases
10856     _tc = read();
10857     for(; tc--; ) AC(); // good luck!
10858 #else
10859     AC(); // good luck!
10860 #endif
10861
10862     return 0;
10863 }
10864
10865 // ----- End of Template -----
10866
10867
10868
10869
10870 constant maxn = 1E6+8;
10871 constant maxm = 2E6+8;
10872
10873 _n, x, y, z;
10874 vector<p> g[maxn];
10875 _p a[maxm];
10876 bool ok;
10877
10878 void dfs(_ x, _ fa, _ tp, _ vl);
10879 inline void AC() {
10880
10881     n = read();
10882     for(_ i = 1; i <= n; ++i) g[i].clear();
10883     for(_ i = 0; i <= maxm-1; ++i) a[i] = mp(-1, -1);
10884     for(_ i = 1; i <= n-1; ++i) {
10885         read({x, y, z});
10886         g[x].push_back(mp(y, z));
10887         g[y].push_back(mp(x, z));
10888     }
10889     ok = false;
10890     for(_ i = 1; i <= n; ++i) dfs(i, -1, i, 0);
10891     if(!ok) writeln(-1);
10892
10893     return;
10894 }
10895
10896 void dfs(_ x, _ fa, _ tp, _ vl) {
10897     if(ok) return;
10898     if(x > tp) {
10899         if(!~a[vl].first) a[vl] = mp(x, tp);
10900         else {
10901             ok = true;
10902             if(a[vl].first > a[vl].second) swap(a[vl].first, a[vl].second);
10903             if(x > tp) swap(x, tp);
10904             writeln({a[vl].first, a[vl].second, x, tp});
10905             return;
10906         }
10907     }
10908     for(_p i : g[x]) if(i.first != fa) {
10909         dfs(i.first, x, tp, vl^i.second);
10910         if(ok) return;
10911     }
10912     return;

```

```

10913 }
10914
10915 //THREEPC
10916 #include <bits/stdc++.h>
10917
10918 using namespace std;
10919
10920 template <typename T>
10921 struct SegmentTree{
10922     int n = 0;
10923     vector<T> tree;
10924
10925     T neutral_element = numeric_limits<T>().max();
10926
10927     SegmentTree(){};
10928     SegmentTree(int _n){
10929         n = _n;
10930         tree.assign(n * 4 + 5, neutral_element);
10931     }
10932
10933     inline T combine(T lf, T rg){
10934         return min(lf, rg);
10935     }
10936
10937     inline void update(int v, int tl, int tr, int pos, T val){
10938         if (tl == tr){
10939             tree[v] = val;
10940             return;
10941         }
10942         int tm = (tl + tr) >> 1;
10943         if (pos <= tm){
10944             update(v << 1, tl, tm, pos, val);
10945         } else {
10946             update(v << 1 | 1, tm + 1, tr, pos, val);
10947         }
10948         tree[v] = combine(tree[v << 1], tree[v << 1 | 1]);
10949     }
10950
10951     inline void update(int pos, T val){
10952         update(1, 0, n - 1, pos, val);
10953     }
10954
10955     inline int get(int v, int tl, int tr, T val){
10956         if (tl == tr) return tl;
10957         int tm = (tl + tr) >> 1;
10958         if (tree[v << 1] <= val){
10959             return get(v << 1, tl, tm, val);
10960         } else {
10961             return get(v << 1 | 1, tm + 1, tr, val);
10962         }
10963     }
10964
10965     inline int get(T val){
10966         if (tree[1] > val) return -1;
10967         return get(1, 0, n - 1, val);
10968     }
10969 };
10970
10971
10972 int find_longest(const vector<int> &a, const vector<int> &b, const vector<int> &c){
10973     int n = (int)a.size();
10974
10975     int ans = 0;
10976     vector<long long> x(n), y(n);
10977     for (int i = 0; i < n; i++){
10978         x[i] = a[i] - b[i];
10979         y[i] = a[i] - c[i];
10980         if (i > 0){
10981             x[i] += x[i - 1];

```

```

10982         y[i] += y[i - 1];
10983     }
10984
10985     if (x[i] >= 0 && y[i] >= 0) ans = max(ans, i + 1);
10986 }
10987
10988 vector<int> perm(n);
10989 iota(perm.begin(), perm.end(), 0);
10990
10991 sort(perm.begin(), perm.end(), [&](int i, int j){
10992     return make_tuple(x[i], y[i], i) < make_tuple(x[j], y[j], j);
10993 });
10994
10995 SegmentTree<long long> st(n);
10996
10997 for (int i : perm){
10998     int j = st.get(y[i]);
10999     if (j != -1 && j < i){
11000         ans = max(ans, i - j);
11001     }
11002     st.update(i, y[i]);
11003 }
11004
11005 return ans;
11006 }
11007
11008 void test_case(){
11009     int n;
11010     cin >> n;
11011
11012     vector<int> a(n), b(n), c(n);
11013     for (int i = 0; i < n; i++) cin >> a[i];
11014     for (int i = 0; i < n; i++) cin >> b[i];
11015     for (int i = 0; i < n; i++) cin >> c[i];
11016
11017     cout << find_longest(a, b, c) << ' ';
11018     cout << find_longest(b, c, a) << ' ';
11019     cout << find_longest(c, a, b) << endl;
11020 }
11021
11022 int main(){
11023     ios_base::sync_with_stdio(false);
11024
11025     int T;
11026     cin >> T;
11027
11028     while (T--){
11029         test_case();
11030     }
11031
11032     return 0;
11033 }
11034
11035 //MATPAIN80
11036 #include <bits/stdc++.h>
11037 #include <ext/pb_ds/assoc_container.hpp>
11038 #include <ext/pb_ds/tree_policy.hpp>
11039 #define PRE(x,p) cout<<setprecision(x)<<p;
11040 #define pb push_back
11041 #define mp make_pair
11042 #define f first
11043 #define s second
11044 #define pi 3.14159265358979
11045 #define mod (1ll)(1e9 + 7)
11046 #define endl "\n"
11047 #define high 1e18
11048 #define low -1e18
11049 #define ll long long int
11050 #define ld long double

```

```

11051 #define mem(x, val)    memset(x, 0, sizeof(x));
11052 #define rep(i, l, r)   for(ll i=l; i<=r; i++)
11053 #define p(a)           for(auto i:a) cout<<i<<' '; cout<<endl;
11054 #define vll            vector<ll>
11055 #define vb             vector<bool>
11056 #define vp11           vector<pair<ll, ll>>
11057 #define vi             vector<int>
11058 #define vpi            vector<pair<int, int>>
11059 #define vv11           vector<vector<ll>>
11060 #define vvi            vector<vector<int>>
11061 #define vv11           vector<vector<vector<ll>>>
11062 #define pll            pair<ll, ll>
11063 #define vs             vector<string>
11064 #define vvp11          vector<vector<pair<ll, ll>>>
11065 #define vvp11          vector<vector<pair<int, int>>>
11066 #define vp11           vector<pair<int, int>>
11067 #define sz(a)          (ll)a.size()
11068 #define po(x)          (ll)(1ll<<x)
11069 #define all(x)          begin(x), end(x)
11070 #define speed          ios_base::sync_with_stdio(false); cin.tie(NULL); cout.tie(NULL);
11071 #define yes            {cout<<"YES"<<endl; return;}
11072 #define no             {cout<<"NO"<<endl; return;}
11073 #define ok             cout<<"ok"<<endl;
11074 #define ordered_set    tree<int, null_type, less<int>,
rb_tree_tag, tree_order_statistics_node_update>

11075
11076
11077 using namespace std;
11078 using namespace __gnu_pbds;
11079
11080 void showa(ll a[], ll n) { for(ll i=1; i<=n; i++) cout<<a[i]<<' '; cout<<endl; }
11081 ll ison(ll w, ll i) {return w&(1ll<<i);}
11082 void amax(ll &a, ll b){ a=max(a,b); }
11083 void amin(ll &a, ll b){ a=min(a,b); }
11084 void modadd(ll &a, ll b) {a=((a%mod)+(b%mod))%mod;}
11085 void modsub(ll &a, ll b) {a=((a%mod)-(b%mod)+mod)%mod;}
11086 void modmul(ll &a, ll b) {a=((a%mod)*(b%mod))%mod;}
11087
11088 #ifndef ONLINE_JUDGE
11089 #define debug(x) cerr << #x <<" "; _print(x); cerr << endl;
11090 #else
11091 #define debug(x)
11092 #endif
11093
11094 void _print(ll t) {cerr << t<<' ';}
11095 void _print(int t) {cerr << t<<' ';}
11096 void _print(string t) {cerr << t<<' ';}
11097 void _print(char t) {cerr << t<<' ';}
11098 void _print(ld t) {cerr << t<<' ';}
11099 void _print(double t) {cerr << t<<' ';}
11100 template<class T, class V> void _print(pair<T, V> p);
11101 template<class T> void _print(vector<T> v);
11102 template<class T> void _print(vector<T> v);
11103 template<class T> void _print(set<T> v);
11104 template<class T, class V> void _print(map<T, V> v);
11105 template<class T> void _print(multiset<T> v);
11106 template<class T, class V> void _print(pair<T, V> p) {cerr << "{"; _print(p.f); cerr <<
", "; _print(p.s); cerr << "}";}
11107 template<class T> void _print(vector<T> v) {cerr << "["; for (T i : v) {_print(i); cerr
<< " ";} cerr << "];"}
11108 template<class T> void _print(set<T> v) {cerr << "["; for (T i : v) {_print(i); cerr <<
" ";} cerr << "];"}
11109 template<class T> void _print(multiset<T> v) {cerr << "["; for (T i : v) {_print(i);
cerr << " ";} cerr << "];"}
11110 template<class T, class V> void _print(map<T, V> v) {cerr << "["; for (auto i : v) {
_print(i); cerr << " ";} cerr << "];"}
11111
11112 //const ll l=30; //log2(n)
11113 //const ll N=200005;

```

```

11114
11115 ll n,m,sum_k;
11116
11117 ll get(ll x)
11118 {
11119     x%=mod;
11120     ll ans=(x*(x+1))%mod;
11121     modmul(ans,500000004);
11122     return ans;
11123 }
11124 ll getrow(ll x) //get value of xth row
11125 {
11126     ll ans= get(x*m);
11127     modsub( ans, get ((x-1)*m));
11128     return ans;
11129 }
11130 ll getcol(ll x) //get value of xth row
11131 {
11132     ll ans=n;
11133     ll here=(2*x)%mod;
11134     modadd(here, (n-1)*m);
11135     modmul(ans,here);
11136     modmul(ans,500000004);
11137     return ans;
11138 }
11139 void check(ll x , ll l , ll r){
11140     assert(x>=l && x<=r);
11141 }
11142
11143 ll rec(vvll vec)
11144 {
11145     ll ans=0;
11146     for(vll v:vec)
11147     {
11148         ll here=1;
11149         for(ll x:v) modmul(here,x);
11150         modadd(ans,here);
11151     }
11152     return ans;
11153 }
11154 void solve()
11155 {
11156     ll k;
11157     assert(cin>>n>>m>>k);
11158     sum_k+=k;
11159     check(n,1,1e9);
11160     check(m,1,1e9);
11161     check(k,1,2e5);
11162     check(sum_k,1,2e5);
11163     map<ll,ll> row,col;
11164     rep(i,1,k)
11165     {
11166         ll type,x,c;
11167         assert(cin>>type>>x>>c);
11168         check(type,0,1);
11169         check(c,0,1e9);
11170         if(type==0)
11171         {
11172             check(x,1,n);
11173             assert(row.count(x)==0);
11174             row[x]=c;
11175         }
11176         if(type==1)
11177         {
11178             check(x,1,m);
11179             assert(col.count(x)==0);
11180             col[x]=c;
11181         }
11182     }

```

```

11183     ll yy=0;
11184     ll cc_yy=0;
11185
11186     for(auto it:col)
11187     {
11188         modadd(yy,it.s);
11189         modadd(cc_yy, it.f*it.s);
11190     }
11191
11192     ll ans=get(n*m);
11193     for(auto it:row) modadd(ans,getrow(it.f) * (it.s-1+mod)%mod);
11194     for(auto it:col) modadd(ans,getcol(it.f) * (it.s-1+mod)%mod);
11195
11196     ll tot_c=col.size()%mod;
11197     ll col_sum=0;
11198     for(auto it:col) modadd(col_sum,it.f);
11199     for(auto it:row)
11200     {
11201         ll r=it.f;
11202         ll x=it.s;
11203
11204         ll here= ((r-1)*m)%mod;
11205         modmul(here,x);
11206         modmul(here,yy);
11207
11208         ll toadd=x;
11209         modmul(toadd, cc_yy);
11210         modadd(here,toadd);
11211         modadd(ans,here);
11212
11213         ll tosub1=rec({{m,r-1,x,tot_c},{x,col_sum}});
11214         ll tosub2=rec({{m,r-1,yy},{cc_yy}});
11215         modsub(tosub2, m*((r-1)*tot_c)%mod);
11216         modsub(tosub2, col_sum);
11217
11218         modsub(ans,tosub1);
11219         modsub(ans,tosub2);
11220
11221     }
11222     cout<<ans<<endl;
11223
11224 }
11225
11226 signed main()
11227 {
11228     #ifndef ONLINE_JUDGE
11229         freopen("input_5.in", "r", stdin);
11230         freopen("output_5.out", "w", stdout);
11231     #endif
11232     speed
11233
11234     ll t=1;
11235     assert(cin>>t);
11236     check(t,1,100);
11237
11238     for(ll test=1;test<=t;test++)
11239     {
11240         solve();
11241     }
11242     return 0;
11243 }
11244
11245 //EATROCK
11246 #include "bits/stdc++.h"
11247 // #pragma GCC optimize("O3,unroll-loops")
11248 // #pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
11249 using namespace std;
11250 using ll = long long int;
11251 mt19937_64 rng(chrono::high_resolution_clock::now().time_since_epoch().count());

```

```

11252
11253 /**
11254  * Integers modulo p, where p is a prime
11255  * Source: Aeren (modified from tourist?)
11256  *      Modmul for 64-bit mod from kactl:ModMulLL
11257  * Works with p < 7.2e18 with x87 80-bit long double, and p < 2^52 ~ 4.5e12 with 64-bit
11258  */
11259 template<typename T>
11260 struct Z_p{
11261     using Type = typename decay<decltype(T::value)>::type;
11262     static vector<Type> MOD_INV;
11263     constexpr Z_p(): value(){ }
11264     template<typename U> Z_p(const U &x){ value = normalize(x); }
11265     template<typename U> static Type normalize(const U &x){
11266         Type v;
11267         if(-mod() <= x && x < mod()) v = static_cast<Type>(x);
11268         else v = static_cast<Type>(x % mod());
11269         if(v < 0) v += mod();
11270         return v;
11271     }
11272     const Type& operator()() const{ return value; }
11273     template<typename U> explicit operator U() const{ return static_cast<U>(value); }
11274     constexpr static Type mod(){ return T::value; }
11275     Z_p &operator+=(const Z_p &otr){ if((value += otr.value) >= mod()) value -= mod();
11276     return *this; }
11277     Z_p &operator-=(const Z_p &otr){ if((value -= otr.value) < 0) value += mod(); return
11278     *this; }
11279     template<typename U> Z_p &operator+=(const U &otr){ return *this += Z_p(otr); }
11280     template<typename U> Z_p &operator-=(const U &otr){ return *this -= Z_p(otr); }
11281     Z_p &operator++(){ return *this += 1; }
11282     Z_p &operator--(){ return *this -= 1; }
11283     Z_p operator++(int){ Z_p result(*this); *this += 1; return result; }
11284     Z_p operator--(int){ Z_p result(*this); *this -= 1; return result; }
11285     Z_p operator-() const{ return Z_p(-value); }
11286     template<typename U = T>
11287     typename enable_if<is_same<typename Z_p<U>::Type, int>::value, Z_p>::type &operator
11288     *=(const Z_p& rhs){
11289         #ifdef WIN32
11290         uint64_t x = static_cast<uint64_t>(value) * static_cast<uint64_t>(rhs.value);
11291         uint32_t xh = static_cast<uint32_t>(x >> 32), xl = static_cast<uint32_t>(x), d,
11292         m;
11293         asm(
11294             "divl %4; \n\t"
11295             : "=a" (d), "=d" (m)
11296             : "d" (xh), "a" (xl), "r" (mod())
11297         );
11298         value = m;
11299         #else
11300         value = normalize(static_cast<int64_t>(value) * static_cast<int64_t>(rhs.value));
11301         #endif
11302         return *this;
11303     }
11304     template<typename U = T>
11305     typename enable_if<is_same<typename Z_p<U>::Type, int64_t>::value, Z_p>::type &
11306     operator*=(const Z_p &rhs){
11307         uint64_t ret = static_cast<uint64_t>(value) * static_cast<uint64_t>(rhs.value) -
11308         static_cast<uint64_t>(mod()) * static_cast<uint64_t>(1.L / static_cast<uint64_t>
11309         >(mod()) * static_cast<uint64_t>(value) * static_cast<uint64_t>(rhs.value));
11310         value = normalize(static_cast<int64_t>(ret + static_cast<uint64_t>(mod()) * (ret
11311         < 0) - static_cast<uint64_t>(mod()) * (ret >= static_cast<uint64_t>(mod()))));
11312         return *this;
11313     }
11314     template<typename U = T>
11315     typename enable_if<!is_integral<typename Z_p<U>::Type>::value, Z_p>::type &operator
11316     *=(const Z_p &rhs){
11317         value = normalize(value * rhs.value);
11318         return *this;
11319     }
11320     template<typename U>

```

```

11312     Z_p &operator^=(U e){
11313         if(e < 0) *this = 1 / *this, e = -e;
11314         Z_p res = 1;
11315         for(; e; *this *= *this, e >= 1) if(e & 1) res *= *this;
11316         return *this = res;
11317     }
11318     template<typename U>
11319     Z_p operator^(U e) const{
11320         return Z_p(*this) ^= e;
11321     }
11322     Z_p &operator/=(const Z_p &otr){
11323         Type a = otr.value, m = mod(), u = 0, v = 1;
11324         if(a < (int)MOD_INV.size()) return *this *= MOD_INV[a];
11325         while(a){
11326             Type t = m / a;
11327             m -= t * a; swap(a, m);
11328             u -= t * v; swap(u, v);
11329         }
11330         assert(m == 1);
11331         return *this *= u;
11332     }
11333     template<typename U> friend const Z_p<U> &abs(const Z_p<U> &v){ return v; }
11334     Type value;
11335 };
11336 template<typename T> bool operator==(const Z_p<T> &lhs, const Z_p<T> &rhs){ return lhs.
value == rhs.value; }
11337 template<typename T, typename U, typename enable_if<is_integral<U>::value>::type* =
nullptr> bool operator==(const Z_p<T>& lhs, U rhs){ return lhs == Z_p<T>(rhs); }
11338 template<typename T, typename U, typename enable_if<is_integral<U>::value>::type* =
nullptr> bool operator==(U lhs, const Z_p<T> &rhs){ return Z_p<T>(lhs) == rhs; }
11339 template<typename T> bool operator!=(const Z_p<T> &lhs, const Z_p<T> &rhs){ return !(lhs
== rhs); }
11340 template<typename T, typename U, typename enable_if<is_integral<U>::value>::type* =
nullptr> bool operator!=(const Z_p<T> &lhs, U rhs){ return !(lhs == rhs); }
11341 template<typename T, typename U, typename enable_if<is_integral<U>::value>::type* =
nullptr> bool operator!=(U lhs, const Z_p<T> &rhs){ return !(lhs == rhs); }
11342 template<typename T> bool operator<(const Z_p<T> &lhs, const Z_p<T> &rhs){ return lhs.
value < rhs.value; }
11343 template<typename T> bool operator>(const Z_p<T> &lhs, const Z_p<T> &rhs){ return lhs.
value > rhs.value; }
11344 template<typename T> bool operator<=(const Z_p<T> &lhs, const Z_p<T> &rhs){ return lhs.
value <= rhs.value; }
11345 template<typename T> bool operator>=(const Z_p<T> &lhs, const Z_p<T> &rhs){ return lhs.
value >= rhs.value; }
11346 template<typename T> Z_p<T> operator+(const Z_p<T> &lhs, const Z_p<T> &rhs){ return Z_p<
T>(lhs) += rhs; }
11347 template<typename T, typename U, typename enable_if<is_integral<U>::value>::type* =
nullptr> Z_p<T> operator+(const Z_p<T> &lhs, U rhs){ return Z_p<T>(lhs) += rhs; }
11348 template<typename T, typename U, typename enable_if<is_integral<U>::value>::type* =
nullptr> Z_p<T> operator+(U lhs, const Z_p<T> &rhs){ return Z_p<T>(lhs) += rhs; }
11349 template<typename T> Z_p<T> operator-(const Z_p<T> &lhs, const Z_p<T> &rhs){ return Z_p<
T>(lhs) -= rhs; }
11350 template<typename T, typename U, typename enable_if<is_integral<U>::value>::type* =
nullptr> Z_p<T> operator-(const Z_p<T>& lhs, U rhs){ return Z_p<T>(lhs) -= rhs; }
11351 template<typename T, typename U, typename enable_if<is_integral<U>::value>::type* =
nullptr> Z_p<T> operator-(U lhs, const Z_p<T> &rhs){ return Z_p<T>(lhs) -= rhs; }
11352 template<typename T> Z_p<T> operator*(const Z_p<T> &lhs, const Z_p<T> &rhs){ return Z_p<
T>(lhs) *= rhs; }
11353 template<typename T, typename U, typename enable_if<is_integral<U>::value>::type* =
nullptr> Z_p<T> operator*(const Z_p<T>& lhs, U rhs){ return Z_p<T>(lhs) *= rhs; }
11354 template<typename T, typename U, typename enable_if<is_integral<U>::value>::type* =
nullptr> Z_p<T> operator*(U lhs, const Z_p<T> &rhs){ return Z_p<T>(lhs) *= rhs; }
11355 template<typename T> Z_p<T> operator/(const Z_p<T> &lhs, const Z_p<T> &rhs) { return Z_p
<T>(lhs) /= rhs; }
11356 template<typename T, typename U, typename enable_if<is_integral<U>::value>::type* =
nullptr> Z_p<T> operator/(const Z_p<T>& lhs, U rhs) { return Z_p<T>(lhs) /= rhs; }
11357 template<typename T, typename U, typename enable_if<is_integral<U>::value>::type* =
nullptr> Z_p<T> operator/(U lhs, const Z_p<T> &rhs) { return Z_p<T>(lhs) /= rhs; }
11358 template<typename T> istream &operator>>(istream &in, Z_p<T> &number){

```



```

11359     typename common_type<typename Z_p<T>::Type, int64_t>::type x;
11360     in >> x;
11361     number.value = Z_p<T>::normalize(x);
11362     return in;
11363 }
11364 template<typename T> ostream &operator<<(ostream &out, const Z_p<T> &number){ return out
    << number(); }

11365
11366 /*
11367 using ModType = int;
11368 struct VarMod{ static ModType value; };
11369 ModType VarMod::value;
11370 ModType &mod = VarMod::value;
11371 using Zp = Z_p<VarMod>;
11372 */
11373
11374 constexpr int mod = 1e9 + 7; // 1000000007
11375 // constexpr int mod = (119 << 23) + 1; // 998244353
11376 // constexpr int mod = 1e9 + 9; // 1000000009
11377 using Zp = Z_p<integral_constant<decay<decltype(mod)>::type, mod>>;
11378
11379 template<typename T> vector<typename Z_p<T>::Type> Z_p<T>::MOD_INV;
11380 template<typename T = integral_constant<decay<decltype(mod)>::type, mod>>
11381 void precalc_inverse(int SZ){
11382     auto &inv = Z_p<T>::MOD_INV;
11383     if(inv.empty()) inv.assign(2, 1);
11384     for(; inv.size() <= SZ; ) inv.push_back((mod - 1LL * mod / (int)inv.size() * inv[mod
        % (int)inv.size()]) % mod);
11385 }
11386
11387 template<typename T>
11388 vector<T> precalc_power(T base, int SZ){
11389     vector<T> res(SZ + 1, 1);
11390     for(auto i = 1; i <= SZ; ++ i) res[i] = res[i - 1] * base;
11391     return res;
11392 }
11393
11394 template<typename T>
11395 vector<T> precalc_factorial(int SZ){
11396     vector<T> res(SZ + 1, 1); res[0] = 1;
11397     for(auto i = 1; i <= SZ; ++ i) res[i] = res[i - 1] * i;
11398     return res;
11399 }
11400
11401 struct Data {
11402     Zp powsum = 0, pospowsum = 0;
11403 }unit;
11404
11405 /**
11406  * Point-update Segment Tree
11407  * Source: kactl
11408  * Description: Iterative point-update segment tree, ranges are half-open i.e [L, R).
11409  *               f is any associative function.
11410  * Time: O(logn) update/query
11411  */
11412
11413 struct SegTree {
11414     using T = Data;
11415     T f(T a, T b) {
11416         a.powsum += b.powsum;
11417         a.pospowsum += b.pospowsum;
11418         return a;
11419     }
11420     vector<T> s; int n;
11421     SegTree(int _n = 0, T def = unit) : s(2*_n, def), n(_n) {}
11422     void update(int pos, T val) {
11423         for (s[pos += n] = val; pos /= 2;)
11424             s[pos] = f(s[pos * 2], s[pos * 2 + 1]);
11425     }

```

```

11426     T query(int b, int e) {
11427         T ra = unit, rb = unit;
11428         for (b += n, e += n; b < e; b /= 2, e /= 2) {
11429             if (b % 2) ra = f(ra, s[b++]);
11430             if (e % 2) rb = f(s[--e], rb);
11431         }
11432         return f(ra, rb);
11433     }
11434 };
11435
11436 int main()
11437 {
11438     ios::sync_with_stdio(false); cin.tie(0);
11439
11440     int t; cin >> t;
11441     while (t--) {
11442         int n; cin >> n;
11443         vector<int> pos(n), wt(n), where(n+1);
11444         for (int &x : pos) cin >> x;
11445         for (int &x : wt) cin >> x;
11446
11447         for (int i = 0; i < n; ++i) where[wt[i]] = i;
11448
11449         Zp ans = 0;
11450         SegTree seg(n);
11451         for (int i = 1; i <= n; ++i) {
11452             int u = where[i];
11453             /**
11454              * For v > u, (pos[v] - pos[u])*2^(n-i + wt[v]-1)
11455              * 2^(n-i) is constant
11456              * (pos[v] - pos[u]) * 2^(wt[v] - 1)
11457              * = pos[v]*(2 ^ (wt[v] - 1)) - pos[u]*2^(wt[v] - 1)
11458              *
11459              * For v < u, similar
11460              */
11461             auto right = seg.query(u+1, n);
11462             Zp pw = Zp(2) ^ (n - i);
11463             ans += pw * (right.pospowsum - right.powsum*pos[u]);
11464
11465             auto left = seg.query(0, u);
11466             ans += pw * (left.powsum*pos[u] - left.pospowsum);
11467
11468             pw = Zp(2) ^ (i - 1);
11469             Data cur = {pw, pw * pos[u]};
11470             seg.update(u, cur);
11471         }
11472         ans /= Zp(2) ^ n;
11473
11474         cout << ans << '\n';
11475     }
11476 }
11477
11478 //MOUNTAIN
11479 #include <map>
11480 #include <set>
11481 #include <cmath>
11482 #include <ctime>
11483 #include <queue>
11484 #include <stack>
11485 #include <cstdio>
11486 #include <cstdlib>
11487 #include <vector>
11488 #include <cstring>
11489 #include <algorithm>
11490 #include <iostream>
11491 using namespace std;
11492 typedef double db;
11493 typedef long long ll;
11494 typedef unsigned long long ull;

```

```

11495 const int N=1000010;
11496 const int LOGN=28;
11497 const ll TMD=0;
11498 const ll INF=2147483647;
11499 int n,m,q;
11500 int p[N];
11501 pair<ll,ll> qr[N];
11502 vector<int> ans[N];
11503
11504 int main()
11505 {
11506     scanf("%d%d%d",&n,&m,&q);
11507     for(int i=1;i<=q;i++)
11508     {
11509         ll t,sum;
11510         int L=0,R=n+1,M,p;
11511         scanf("%lld",&t);
11512         while(L+1!=R)
11513         {
11514             M=(L+R)>>1;
11515             if((ll)m*(ll)M*(M+1)/2<t) L=M;
11516             else R=M;
11517         }
11518         p=R;sum=(ll)m*(ll)L*(L+1)/2;
11519         for(int j=1;j<=m;j++)
11520         {
11521             sum+=p;
11522             if(sum>=t)
11523             {
11524                 printf("%d %d\n",1,p);
11525                 for(int k=1;k<p;k++) printf("%d ",k==sum-t?m-1:m);
11526                 printf("%d\n",j);
11527                 break;
11528             }
11529         }
11530     }
11531
11532     return 0;
11533 }
11534
11535 //SQRTCBRT
11536 #include <bits/stdc++.h>
11537 using namespace std;
11538 #define ll long long
11539 int main() {
11540     vector<ll> cubes;
11541     for(ll i=1;i<=1010000;i++) cubes.push_back(i*i*i);
11542     ll T;
11543     cin >> T;
11544     while(T--){
11545         ll x;
11546         cin >> x;
11547         ll l=1;
11548         ll r=2e9;
11549         ll ans;
11550         while(l<=r){
11551             ll mid=(r+l)/2;
11552             ll temp=upper_bound(cubes.begin(),cubes.end(),mid*mid)-cubes.begin();
11553             ll curr=mid-temp;
11554             if(curr>=x){
11555                 ans=mid*mid;
11556                 r=mid-1;
11557             }
11558             else{
11559                 l=mid+1;
11560             }
11561         }
11562         cout << ans << endl;
11563     }

```

```

11564     return 0;
11565 }
11566
11567 //KBEAUTIFUL
11568 #include <bits/stdc++.h>
11569 using namespace std;
11570 const int MXN=2000010;
11571 const long long MOD=998244353,INF=1000000000;
11572 long long f[MXN],inv[MXN],finv[MXN];
11573 void Initialize(){
11574     f[0]=f[1]=inv[0]=inv[1]=finv[0]=finv[1]=1;
11575     for (int i=2; i<MXN; i++){
11576         f[i]=f[i-1]*i%MOD;
11577         inv[i]=inv[MOD%i]*(MOD-MOD/i)%MOD;
11578         finv[i]=finv[i-1]*inv[i]%MOD;
11579     }
11580 }
11581 long long nCr(int n,int r){
11582     if (n<r) return 0LL;
11583     return f[n]*finv[r]%MOD*finv[n-r]%MOD;
11584 }
11585 long long nHr(int n,int r){
11586     return nCr(n+r-1,r);
11587 }
11588 void solve(){
11589     int n,m,k;
11590     cin>>n>>m>>k;
11591     long long a[n+1];
11592     for (int i=1; i<=n; i++) cin>>a[i];
11593     long long mx[k+1];
11594     for (int i=1; i<=k; i++){
11595         mx[i]=0;
11596         for (int j=i; j<=n; j+=k){
11597             mx[i]=max(mx[i],a[j]);
11598         }
11599         for (int j=i; j<=n; j+=k){
11600             m-=min(INF,mx[i]-a[j]);
11601             if (m<0){
11602                 cout<<"0\n";
11603                 return;
11604             }
11605         }
11606     }
11607     long long sum[m/(n/k)+1];
11608     for (int i=0; i<=m/(n/k); i++){
11609         if (i>0) sum[i]=sum[i-1];
11610         else sum[i]=0;
11611         sum[i]+=nHr(k-n%k,i);
11612         sum[i]%=MOD;
11613     }
11614     long long ans=0;
11615     if (n%k==0){
11616         int mxCnt=m/(n/k);
11617         cout<<sum[mxCnt]<<"\n";
11618         return;
11619     }
11620     for (int cntBig=0; cntBig<=m/((n+k-1)/k); cntBig++){
11621         int mxSmall=(m-(n+k-1)/k*cntBig)/(n/k);
11622         ans+=nHr(n%k,cntBig)*sum[mxSmall];
11623         ans%=MOD;
11624     }
11625     cout<<ans<<"\n";
11626 }
11627 int main(){
11628     ios_base::sync_with_stdio(0); cin.tie(0);
11629     int T=1;
11630     cin>>T;
11631     Initialize();
11632     while (T-->0) solve();

```

```

11633 }
11634
11635 //SPLITORDEC
11636 //Utkarsh.25dec
11637 #include <iostream>
11638 #include <cstdio>
11639 #include <cstdlib>
11640 #include <algorithm>
11641 #include <cmath>
11642 #include <vector>
11643 #include <set>
11644 #include <map>
11645 #include <unordered_set>
11646 #include <unordered_map>
11647 #include <queue>
11648 #include <ctime>
11649 #include <cassert>
11650 #include <complex>
11651 #include <string>
11652 #include <cstring>
11653 #include <chrono>
11654 #include <random>
11655 #include <bitset>
11656 #include <array>
11657 #define ll long long int
11658 #define pb push_back
11659 #define mp make_pair
11660 #define mod 1000000007
11661 #define vl vector<ll>
11662 #define all(c) (c).begin(), (c).end()
11663 using namespace std;
11664 ll power(ll a,ll b) {ll res=1;a%=mod; assert(b>=0); for(;b;b>>=1){if(b&1)res=res*a%mod;a
=a*a%mod;}return res;}
11665 ll modInverse(ll a){return power(a,mod-2);}
11666 const int N=500023;
11667 bool vis[N];
11668 vector<int> adj[N];
11669 long long readInt(long long l,long long r,char endd){
11670     long long x=0;
11671     int cnt=0;
11672     int fi=-1;
11673     bool is_neg=false;
11674     while(true){
11675         char g=getchar();
11676         if(g=='-'){
11677             assert(fi==-1);
11678             is_neg=true;
11679             continue;
11680         }
11681         if('0'<=g && g<='9'){
11682             x*=10;
11683             x+=g-'0';
11684             if(cnt==0){
11685                 fi=g-'0';
11686             }
11687             cnt++;
11688             assert(fi!=0 || cnt==1);
11689             assert(fi!=0 || is_neg==false);
11690
11691             assert(!(cnt>19 || ( cnt==19 && fi>1) ));
11692         } else if(g==endd){
11693             if(is_neg){
11694                 x= -x;
11695             }
11696
11697             if(!(l <= x && x <= r))
11698                 {
11699                     cerr << l << ' ' << r << ' ' << x << '\n';
11700                     assert(1 == 0);

```

```

11701         }
11702
11703         return x;
11704     } else {
11705         assert(false);
11706     }
11707 }
11708 }
11709 string readString(int l,int r,char endd){
11710     string ret="";
11711     int cnt=0;
11712     while(true){
11713         char g=getchar();
11714         assert(g!=-1);
11715         if(g==endd){
11716             break;
11717         }
11718         cnt++;
11719         ret+=g;
11720     }
11721     assert(l<=cnt && cnt<=r);
11722     return ret;
11723 }
11724 long long readIntSp(long long l,long long r){
11725     return readInt(l,r,' ');
11726 }
11727 long long readIntLn(long long l,long long r){
11728     return readInt(l,r,'\n');
11729 }
11730 string readStringLn(int l,int r){
11731     return readString(l,r,'\n');
11732 }
11733 string readStringSp(int l,int r){
11734     return readString(l,r,' ');
11735 }
11736 int sumN=0;
11737 void solve()
11738 {
11739     int n=readInt(1,100000,'\n');
11740     sumN+=n;
11741     assert(sumN<=300000);
11742     int A[n+1];
11743     int even=0,one=0;
11744     for(int i=1;i<=n;i++){
11745         {
11746             if(i==n)
11747                 A[i]=readInt(1,1000000000,'\n');
11748             else
11749                 A[i]=readInt(1,1000000000,' ');
11750             if(A[i]%2==0)
11751                 even++;
11752             if(A[i]==1)
11753                 one++;
11754         }
11755         if(even%2==0 && one%2==0)
11756             cout<<"CHEFINA\n";
11757         else
11758             cout<<"CHEF\n";
11759     }
11760 int main()
11761 {
11762     #ifndef ONLINE_JUDGE
11763     freopen("input.txt", "r", stdin);
11764     freopen("output.txt", "w", stdout);
11765     #endif
11766     ios_base::sync_with_stdio(false);
11767     cin.tie(NULL),cout.tie(NULL);
11768     int T=readInt(1,1000,'\n');
11769     while(T--)
```

```

11770         solve();
11771         assert(getchar() == -1);
11772         cerr << "Time : " << 1000 * ((double)clock()) / (double)CLOCKS_PER_SEC << "ms\n";
11773     }
11774
11775     //SUSPERMS
11776     #include <bits/stdc++.h>
11777     using namespace std;
11778
11779     int n, mod;
11780     struct modint
11781     {
11782         int32_t value;
11783         modint() = default;
11784         modint(int32_t value_) : value(value_ % mod) {}
11785         modint(int64_t value_) : value(value_ % mod) {}
11786         inline modint operator+(modint other) const
11787         {
11788             int32_t c = this->value + other.value;
11789             return modint(c >= mod ? c - mod : c);
11790         }
11791         inline modint operator-(modint other) const
11792         {
11793             int32_t c = this->value - other.value;
11794             return modint(c < 0 ? c + mod : c);
11795         }
11796         inline modint operator*(modint other) const
11797         {
11798             int32_t c = (int64_t)this->value * other.value % mod;
11799             return modint(c < 0 ? c + mod : c);
11800         }
11801         inline modint &operator+=(modint other)
11802         {
11803             this->value += other.value;
11804             if (this->value >= mod)
11805                 this->value -= mod;
11806             return *this;
11807         }
11808         inline modint &operator-=(modint other)
11809         {
11810             this->value -= other.value;
11811             if (this->value < 0)
11812                 this->value += mod;
11813             return *this;
11814         }
11815         inline modint &operator*=(modint other)
11816         {
11817             this->value = (int64_t)this->value * other.value % mod;
11818             if (this->value < 0)
11819                 this->value += mod;
11820             return *this;
11821         }
11822         inline modint operator-() const { return modint(this->value ? mod - this->value : 0); }
11823     };
11824     modint pow(int32_t k) const
11825     {
11826         modint x = *this, y = 1;
11827         for (; k; k >>= 1)
11828         {
11829             if (k & 1)
11830                 y *= x;
11831             x *= x;
11832         }
11833         return y;
11834     }
11835     modint inv() const { return pow(mod - 2); } // MOD must be a prime
11836     inline modint operator/(modint other) const { return *this * other.inv(); }
11837     inline modint operator/=(modint other) { return *this *= other.inv(); }
11838     inline bool operator==(modint other) const { return value == other.value; }

```

```

11838     inline bool operator!=(modint other) const { return value != other.value; }
11839     inline bool operator<(modint other) const { return value < other.value; }
11840     inline bool operator>(modint other) const { return value > other.value; }
11841 };
11842 modint operator*(int64_t value, modint n) { return modint(value) * n; }
11843 modint operator*(int32_t value, modint n) { return modint(value) * n; }
11844 istream &operator>>(istream &in, modint &n) { return in >> n.value; }
11845 ostream &operator<<(ostream &out, modint n) { return out << n.value; }
11846 struct combi
11847 {
11848     int n;
11849     vector<modint> facts, finvs, invs;
11850     combi(int _n) : n(_n), facts(_n), finvs(_n), invs(_n)
11851     {
11852         facts[0] = finvs[0] = 1;
11853         invs[1] = 1;
11854         for (int i = 2; i < n; i++)
11855             invs[i] = invs[mod % i] * (-mod / i);
11856         for (int i = 1; i < n; i++)
11857         {
11858             facts[i] = facts[i - 1] * i;
11859             finvs[i] = finvs[i - 1] * invs[i];
11860         }
11861     }
11862     inline modint fact(int n) { return facts[n]; }
11863     inline modint finv(int n) { return finvs[n]; }
11864     inline modint inv(int n) { return invs[n]; }
11865     inline modint ncr(int n, int k) { return n < k or k < 0 ? 0 : facts[n] * finvs[k] *
11866         finvs[n - k]; }
11867     inline modint aranj(int n, int k) { return ncr(n, k) * facts[k]; }
11868 };
11869 struct base
11870 {
11871     double x, y;
11872     base() { x = y = 0; }
11873     base(double x, double y) : x(x), y(y) {}
11874 };
11875 inline base operator+(base a, base b) { return base(a.x + b.x, a.y + b.y); }
11876 inline base operator-(base a, base b) { return base(a.x - b.x, a.y - b.y); }
11877 inline base operator*(base a, base b) { return base(a.x * b.x - a.y * b.y, a.x * b.y + a
11878     .y * b.x); }
11879 inline base conj(base a) { return base(a.x, -a.y); }
11880 int lim = 1;
11881 vector<base> roots = {{0, 0}, {1, 0}};
11882 vector<int> rev = {0, 1};
11883 const double PI = acos(-1.0);
11884 void ensure_base(int p)
11885 {
11886     if (p <= lim)
11887         return;
11888     rev.resize(1 << p);
11889     for (int i = 0; i < (1 << p); i++)
11890         rev[i] = (rev[i >> 1] >> 1) + ((i & 1) << (p - 1));
11891     roots.resize(1 << p);
11892     while (lim < p)
11893     {
11894         double angle = 2 * PI / (1 << (lim + 1));
11895         for (int i = 1 << (lim - 1); i < (1 << lim); i++)
11896         {
11897             roots[i << 1] = roots[i];
11898             double angle_i = angle * (2 * i + 1 - (1 << lim));
11899             roots[(i << 1) + 1] = base(cos(angle_i), sin(angle_i));
11900         }
11901         lim++;
11902     }
11903 }
11904 void fft(vector<base> &a, int n = -1)
11905 {
11906     if (n == -1)

```



```

11905     n = a.size();
11906     assert((n & (n - 1)) == 0);
11907     int zeros = __builtin_ctz(n);
11908     ensure_base(zeros);
11909     int shift = lim - zeros;
11910     for (int i = 0; i < n; i++)
11911         if (i < (rev[i] >> shift))
11912             swap(a[i], a[rev[i] >> shift]);
11913     for (int k = 1; k < n; k <= 1)
11914     {
11915         for (int i = 0; i < n; i += 2 * k)
11916         {
11917             for (int j = 0; j < k; j++)
11918             {
11919                 base z = a[i + j + k] * roots[j + k];
11920                 a[i + j + k] = a[i + j] - z;
11921                 a[i + j] = a[i + j] + z;
11922             }
11923         }
11924     }
11925 }
11926 vector<int> multiply(vector<int> &a, vector<int> &b, int eq = 0)
11927 {
11928     int need = a.size() + b.size() - 1;
11929     int p = 0;
11930     while ((1 << p) < need)
11931         p++;
11932     ensure_base(p);
11933     int sz = 1 << p;
11934     vector<base> A, B;
11935     if (sz > (int)A.size())
11936         A.resize(sz);
11937     for (int i = 0; i < (int)a.size(); i++)
11938     {
11939         int x = (a[i] % mod + mod) % mod;
11940         A[i] = base(x & ((1 << 15) - 1), x >> 15);
11941     }
11942     fill(A.begin() + a.size(), A.begin() + sz, base{0, 0});
11943     fft(A, sz);
11944     if (sz > (int)B.size())
11945         B.resize(sz);
11946     if (eq)
11947         copy(A.begin(), A.begin() + sz, B.begin());
11948     else
11949     {
11950         for (int i = 0; i < (int)b.size(); i++)
11951         {
11952             int x = (b[i] % mod + mod) % mod;
11953             B[i] = base(x & ((1 << 15) - 1), x >> 15);
11954         }
11955         fill(B.begin() + b.size(), B.begin() + sz, base{0, 0});
11956         fft(B, sz);
11957     }
11958     double ratio = 0.25 / sz;
11959     base r2(0, -1), r3(ratio, 0), r4(0, -ratio), r5(0, 1);
11960     for (int i = 0; i <= (sz >> 1); i++)
11961     {
11962         int j = (sz - i) & (sz - 1);
11963         base a1 = (A[i] + conj(A[j])), a2 = (A[i] - conj(A[j])) * r2;
11964         base b1 = (B[i] + conj(B[j])) * r3, b2 = (B[i] - conj(B[j])) * r4;
11965         if (i != j)
11966         {
11967             base c1 = (A[j] + conj(A[i])), c2 = (A[j] - conj(A[i])) * r2;
11968             base d1 = (B[j] + conj(B[i])) * r3, d2 = (B[j] - conj(B[i])) * r4;
11969             A[i] = c1 * d1 + c2 * d2 * r5;
11970             B[i] = c1 * d2 + c2 * d1;
11971         }
11972         A[j] = a1 * b1 + a2 * b2 * r5;
11973         B[j] = a1 * b2 + a2 * b1;

```

```

11974     }
11975     fft(A, sz);
11976     fft(B, sz);
11977     vector<int> res(need);
11978     for (int i = 0; i < need; i++)
11979     {
11980         long long aa = A[i].x + 0.5;
11981         long long bb = B[i].x + 0.5;
11982         long long cc = A[i].y + 0.5;
11983         res[i] = (aa + ((bb % mod) << 15) + ((cc % mod) << 30)) % mod;
11984     }
11985     return res;
11986 }
11987 int main()
11988 {
11989     cin.tie(nullptr)->sync_with_stdio(false);
11990     cin >> n >> mod;
11991     combi C(n + n + 1);
11992     int lim = n / 2;
11993     vector<int> first(lim + 1);
11994     for (int i = 0; i <= lim; ++i)
11995     {
11996         modint rep = C.fact(4 * i) * C.finv(2 * i) * C.finv(2 * i);
11997         first[i] = rep.value;
11998     }
11999     vector<int> second(lim + 1);
12000     for (int i = 0; i <= lim; ++i)
12001     {
12002         modint rep = modint(2).pow(i) * modint(-1).pow(i) * C.finv(i);
12003         second[i] = rep.value;
12004     }
12005     vector<int> sumeven = multiply(first, second);
12006     first = vector<int>(lim + 2);
12007     for (int i = 1; i <= lim + 1; ++i)
12008     {
12009         modint rep = C.fact(4 * i - 2) * C.finv(2 * i - 1) * C.finv(2 * i - 1);
12010         first[i] = rep.value;
12011     }
12012     vector<int> sumodd = multiply(first, second);
12013     for (int i = 1; i <= n; ++i)
12014     {
12015         if (i % 2 == 1)
12016         {
12017             modint rep = C.fact(i) * C.fact(i) * sumodd[(i + 1) / 2];
12018             cout << rep << ' ';
12019         }
12020         else
12021         {
12022             modint rep = C.fact(i) * C.fact(i) * sumeven[i / 2];
12023             cout << rep << ' ';
12024         }
12025     }
12026 }
12027
12028 //MULSUBQ
12029 #include "bits/stdc++.h"
12030 // #pragma GCC optimize("O3,unroll-loops")
12031 // #pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
12032 using namespace std;
12033 using ll = long long int;
12034 mt19937_64 rng(chrono::high_resolution_clock::now().time_since_epoch().count());
12035
12036 struct Node {
12037     using T = ll;
12038     T unit = 0;
12039     T f(T a, T b) { return a+b; }
12040
12041     Node *l = 0, *r = 0;
12042     int lo, hi;

```

```

12043 T madd = 0;
12044 T val = unit;
12045 Node(int _lo,int _hi):lo(_lo),hi(_hi){}
12046 T query(int L, int R) {
12047     if (R <= lo || hi <= L) return unit;
12048     if (L <= lo && hi <= R) return val;
12049     push();
12050     return f(l->query(L, R), r->query(L, R));
12051 }
12052 void add(int L, int R, T x) {
12053     if (R <= lo || hi <= L) return;
12054     if (L <= lo && hi <= R) {
12055         madd += x;
12056         val += (hi-lo)*x;
12057     }
12058     else {
12059         push(), l->add(L, R, x), r->add(L, R, x);
12060         val = f(l->val, r->val);
12061     }
12062 }
12063 void push() {
12064     if (!l) {
12065         int mid = lo + (hi - lo)/2;
12066         l = new Node(lo, mid); r = new Node(mid, hi);
12067     }
12068     if (madd)
12069         l->add(lo,hi,madd), r->add(lo,hi,madd), madd = 0;
12070 }
12071 };
12072
12073 int main()
12074 {
12075     ios::sync_with_stdio(false); cin.tie(0);
12076
12077     int t; cin >> t;
12078     while (t--) {
12079         int n, q; cin >> n >> q;
12080         vector<int> a(n);
12081         for (int &x : a) cin >> x;
12082         vector<vector<array<int, 2>>> queries(n);
12083         for (int i = 0; i < q; ++i) {
12084             int l, r; cin >> l >> r; --l;
12085             queries[l].push_back({r, i});
12086         }
12087         vector<ll> ans(q);
12088
12089         Node *seg = new Node(0, n);
12090         set<array<int, 2>> active;
12091         active.insert({n, 0});
12092         vector<int> next(n+1, n);
12093         for (int i = n-1; i >= 0; --i) {
12094             active.erase({next[a[i]], a[i]});
12095             next[a[i]] = i;
12096             active.insert({next[a[i]], a[i]});
12097
12098             auto it = active.begin();
12099             vector<int> cur;
12100             bool good = true;
12101             int lim = i+1;
12102             while (true) {
12103                 auto [pos, val] = *it;
12104                 if (val == 0) {
12105                     lim = n;
12106                     break;
12107                 }
12108                 cur.push_back(val);
12109                 sort(begin(cur), end(cur));
12110                 for (int j = 0; j+1 < (int)size(cur); ++j) good &= cur[j+1] % cur[j] ==
12111                     0;

```

```

12111         if (!good) {
12112             lim = pos;
12113             break;
12114         }
12115         ++it;
12116     }
12117     seg -> add(i, lim, 1);
12118     for (auto [r, id] : queries[i]) {
12119         ans[id] = seg -> query(i, r);
12120     }
12121 }
12122
12123     for (auto x : ans) cout << x << '\n';
12124 }
12125 }
12126
12127 //SOLVEMORE
12128 #include "bits/stdc++.h"
12129 // #pragma GCC optimize("O3,unroll-loops")
12130 // #pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
12131 using namespace std;
12132 using ll = long long int;
12133 mt19937_64 rng(chrono::high_resolution_clock::now().time_since_epoch().count());
12134
12135 int main()
12136 {
12137     ios::sync_with_stdio(false); cin.tie(0);
12138
12139     int t; cin >> t;
12140     while (t--) {
12141         int n, k; cin >> n >> k;
12142         vector<array<int, 2>> v(n);
12143         for (int i = 0; i < n; ++i) cin >> v[i][0];
12144         for (int i = 0; i < n; ++i) cin >> v[i][1];
12145         sort(begin(v), end(v), [](auto a, auto b) {
12146             return a[0] + a[1] < b[0] + b[1];
12147         });
12148         int ans = 0, cur = 0;
12149         for (int i = 0; i < n; ++i) {
12150             cur += v[i][0] + v[i][1];
12151             if (cur <= k) continue;
12152
12153             ans = i;
12154             int mn1 = INT_MAX, mn2 = INT_MAX;
12155             for (int j = 0; j < n; ++j) {
12156                 if (v[j][0] + v[j][1] <= v[i][0] + v[i][1]) mn1 = min(mn1, cur - v[j][1]);
12157             }
12158             if (mn1 <= k) ans = i+1;
12159
12160             cur -= v[i][0] + v[i][1];
12161             for (int j = i; j < n; ++j) {
12162                 mn2 = min(mn2, cur + v[j][0]);
12163             }
12164             cur += v[i][0] + v[i][1];
12165             if (mn2 <= k) ans = i+1;
12166
12167             break;
12168         }
12169         if (cur <= k) ans = n;
12170         cout << ans << '\n';
12171     }
12172 }
12173
12174 //BITEQU
12175 #include <bits/stdc++.h>
12176 using namespace std;
12177 using ll = long long;
12178

```

```

12179 int main() {
12180     ios::sync_with_stdio(false);
12181     cin.tie(0);
12182     cout.tie(0);
12183     int t;
12184     cin >> t;
12185     ll m = 4294967295;
12186     while (t--){
12187         ll n;
12188         cin >> n;
12189         if (n == 0) cout << 17 << " " << 1 << " " << 2 << " " << 3 << endl;
12190         else if (n == m) cout << 2 << " " << 4 << " " << m - 1 << " " << 1 << endl;
12191         else {
12192             ll d = bitset<32>(n).flip().to_ulong();
12193             if (d <= 2) cout << d + 1 << " " << d + 2 << " " << m << " " << d << endl;
12194             else cout << d - 2 << " " << d - 1 << " " << m << " " << d << endl;
12195         }
12196     }
12197     return 0;
12198 }
12199
12200 //BALSUFF
12201 #include <iostream>
12202 #include <string>
12203 #include <set>
12204 #include <map>
12205 #include <stack>
12206 #include <queue>
12207 #include <vector>
12208 #include <utility>
12209 #include <iomanip>
12210 #include <sstream>
12211 #include <bitset>
12212 #include <cstdlib>
12213 #include <iterator>
12214 #include <algorithm>
12215 #include <cstdio>
12216 #include <cctype>
12217 #include <cmath>
12218 #include <math.h>
12219 #include <ctime>
12220 #include <cstring>
12221 #include <unordered_set>
12222 #include <unordered_map>
12223 #include <cassert>
12224 #define int long long int
12225 #define pb push_back
12226 #define mp make_pair
12227 #define mod 1000000007
12228 #define vl vector<ll>
12229 #define all(c) (c).begin(), (c).end()
12230 using namespace std;
12231
12232 const int N=500023;
12233 bool vis[N];
12234 vector<int> adj[N];
12235 long long readInt(long long l, long long r, char endd){
12236     long long x=0;
12237     int cnt=0;
12238     int fi=-1;
12239     bool is_neg=false;
12240     while(true){
12241         char g=getchar();
12242         if(g=='-'){
12243             assert(fi==-1);
12244             is_neg=true;
12245             continue;
12246         }
12247         if('0'<=g && g<='9'){

```

```

12248         x*=10;
12249         x+=g-'0';
12250         if(cnt==0){
12251             fi=g-'0';
12252         }
12253         cnt++;
12254         assert(fi!=0 || cnt==1);
12255         assert(fi!=0 || is_neg==false);
12256
12257         assert(!(cnt>19 || ( cnt==19 && fi>1) ));
12258     } else if(g==endd){
12259         if(is_neg){
12260             x= -x;
12261         }
12262
12263         if(!(l <= x && x <= r))
12264         {
12265             cerr << l << ' ' << r << ' ' << x << '\n';
12266             assert(1 == 0);
12267         }
12268
12269         return x;
12270     } else {
12271         assert(false);
12272     }
12273 }
12274 }
12275 string readString(int l,int r,char endd){
12276     string ret="";
12277     int cnt=0;
12278     while(true){
12279         char g=getchar();
12280         assert(g!=-1);
12281         if(g==endd){
12282             break;
12283         }
12284         cnt++;
12285         ret+=g;
12286     }
12287     assert(l<=cnt && cnt<=r);
12288     return ret;
12289 }
12290 long long readIntSp(long long l,long long r){
12291     return readInt(l,r,' ');
12292 }
12293 long long readIntLn(long long l,long long r){
12294     return readInt(l,r,'\n');
12295 }
12296 string readStringLn(int l,int r){
12297     return readString(l,r,'\n');
12298 }
12299 string readStringSp(int l,int r){
12300     return readString(l,r,' ');
12301 }
12302 }
12303 bool check(unordered_map<int, int>& freq, int k){
12304     int max_freq = 0;
12305     int min_freq = 1e9;
12306     for(auto it: freq){
12307         max_freq = max(max_freq, it.second);
12308         min_freq = min(min_freq, it.second);
12309     }
12310     return (max_freq - min_freq) <= k;
12311 }
12312
12313 string solution(string s, int k){
12314     string ans = "";
12315     unordered_map<int, int> freq;
12316     for(int i = 0; i < s.size(); i++){

```

```

12317         freq[s[i] - 'a']++;
12318     }
12319     if(!check(freq, k)){
12320         return "-1";
12321     }
12322     for(int i = 0; i<s.length(); i++){
12323         bool flag = false;
12324         for(int j = 0; j < 26; j++){
12325             if(freq.find(j)!=freq.end() && freq[j] > 0){
12326                 freq[j]--;
12327                 if(check(freq, k)){
12328                     ans += (char)(j + 'a');
12329                     flag = true;
12330                     break;
12331                 }
12332                 freq[j]++;
12333             }
12334         }
12335         if(!flag){
12336             return "-1";
12337         }
12338     }
12339     string t = s;
12340     sort(all(t));
12341     if(t == ans){
12342         cerr << 1 << endl;
12343     }
12344     return ans;
12345 }
12346
12347 int sumN = 0;
12348
12349 void solve()
12350 {
12351     int n = readInt(1, 100000, ' ');
12352     int k = readInt(1, n, '\n');
12353     sumN += n;
12354     string s = readStringLn(1, n);
12355     cout << solution(s, k) << '\n';
12356 }
12357
12358 int32_t main()
12359 {
12360     #ifndef ONLINE_JUDGE
12361     freopen("input.txt", "r", stdin);
12362     freopen("output.txt", "w", stdout);
12363     #endif
12364     ios_base::sync_with_stdio(false);
12365     cin.tie(NULL),cout.tie(NULL);
12366     int T=readInt(1,2000,'\n');
12367     while(T--){
12368         solve();
12369     }
12370     assert(getchar()==-1);
12371     cerr << sumN << '\n';
12372     assert(sumN <= 200000);
12373     cerr << "Time : " << 1000 * ((double)clock()) / (double)CLOCKS_PER_SEC << "ms\n";
12374 }
12375
12376 //XSORT
12377 #include <bits/stdc++.h>
12378 #define ll long long
12379 #define int long long
12380 #define fi first
12381 #define se second
12382 #define mat vector<vector<ll>>
12383 using namespace std;
12384 void db() {cout << '\n';}
12385 template <typename T, typename ...U> void db(T a, U ...b) {cout << a << ' ', db(b...);}

```

```

12386 #ifdef Cloud
12387 #define file freopen("input.txt", "r", stdin), freopen("output.txt", "w", stdout)
12388 #else
12389 #define file ios::sync_with_stdio(false); cin.tie(0)
12390 #endif
12391 auto SEED = chrono::steady_clock::now().time_since_epoch().count();
12392 mt19937 rng(SEED);
12393 const int N = 1e5 + 1, mod = 998244353, inf = 111 << 60;
12394 int a[N];
12395 vector<pair<int, int>> v;
12396 void work(int l, int r){
12397     if (l == r) return;
12398     int mid = l + r >> 1;
12399     work(l, mid), work(mid + 1, r);
12400     for (int i = 0; i < r - mid; i++) {
12401         if (a[l + i] == a[mid + 1 + i]) continue;
12402         a[l + i] = a[mid + 1 + i] = a[l + i] ^ a[mid + 1 + i];
12403         v.push_back({l + i, mid + 1 + i});
12404     }
12405 }
12406 int f(int n){
12407     return __lg(n) * n / 2;
12408 }
12409 void solve(){
12410     int n;
12411     cin >> n;
12412     vector<pair<int, int>> ans;
12413     for (int i = 0; i < n; i++) cin >> a[i];
12414     int k = 1;
12415     while (k * 2 < n) k *= 2;
12416     int mid = k * 2 - n;
12417     for (int i = 0; i < n - 1 - i; i++){
12418         if (a[i] == a[n - 1 - i]) continue;
12419         ans.push_back({i + 1, n - i});
12420         a[i] = a[n - 1 - i] = a[i] ^ a[n - 1 - i];
12421     }
12422     work(0, k - 1);
12423     work(n - k, n - 1);
12424     for (auto i : v){
12425         if (a[0] > a[n - 1]){
12426             ans.push_back({n - i.fi, n - i.se});
12427         }
12428         else{
12429             ans.push_back({i.fi + 1, i.se + 1});
12430         }
12431     }
12432     v.clear();
12433     cout << ans.size() << '\n';
12434     for (auto i : ans) cout << i.fi << ' ' << i.se << '\n';
12435 }
12436 signed main(){
12437     file;
12438     int t;
12439     cin >> t;
12440     while (t--) solve();
12441 }
12442
12443 //MYPROBLEM
12444 #include<bits/stdc++.h>
12445 // #pragma GCC optimize("O2")
12446 using namespace std;
12447 using ll = long long;
12448 using ld = long double;
12449 #define pb push_back
12450 #define mp make_pair
12451 #define fi first
12452 #define se second
12453 #define sz(x) (int)x.size()
12454 // #define endl '\n'

```



```

12455     const int mod = 1e9 + 7;
12456     const int inf = 2e9 + 5;
12457     const ll linf = 9e18 + 5;
12458
12459
12460     int n;
12461     int k;
12462
12463     void init() {
12464     }
12465
12466     void input() {
12467         cin >> n >> k;
12468     }
12469
12470     void solve() {
12471         string s = to_string(n);
12472
12473         int ans = 0;
12474         if (n == 1000 * 1000 * 1000) {
12475             ans = 1111111111;
12476         }
12477         else {
12478             for (int i = 0; i < sz(s); i++) {
12479                 ans = ans * 10 + 9;
12480             }
12481         }
12482
12483         set<int> digits;
12484         int now = 0;
12485         for (int i = 0; i < sz(s); i++) {
12486             if (sz(digits) > k) {
12487                 break;
12488             }
12489
12490             int d = s[i] - '0';
12491             if (d == 9) {
12492                 now *= 10;
12493                 now += d;
12494                 digits.insert(d);
12495                 continue;
12496             }
12497
12498
12499             int fore = 0;
12500             int exten = 0;
12501             if (sz(digits) == k) {
12502                 if (digits.upper_bound(d) == digits.end()) {
12503                     now *= 10;
12504                     now += d;
12505                     digits.insert(d);
12506                     continue;
12507                 }
12508
12509                 fore = *digits.upper_bound(d);
12510                 exten = min(fore, *digits.begin());
12511             }
12512             else if (sz(digits) == k - 1) {
12513                 if (digits.find(d + 1) != digits.end()) {
12514                     fore = d + 1;
12515                     exten = 0;
12516                 }
12517                 else {
12518                     fore = d + 1;
12519
12520                     exten = fore;
12521                     if (!digits.empty()) {
12522                         exten = min(exten, *digits.begin());
12523                     }

```

```

12524     }
12525 }
12526 else {
12527     fore = d + 1;
12528     exten = 0;
12529 }
12530
12531
12532     int ans2 = now;
12533
12534     ans2 = ans2 * 10 + fore;
12535 //     cout << "         " << ans2 << endl;
12536     for (int j = i + 1; j < sz(s); j++) {
12537         ans2 = ans2 * 10 + exten;
12538 //         cout << "         " << i << ' ' << j << endl;
12539     }
12540
12541     ans = ans2;
12542 //cout << ans << ' ' << ans2 << ' ' << now << ' ' << fore << ' ' << exten <<
endl;
12543
12544     now *= 10;
12545     now += d;
12546     digits.insert(d);
12547 }
12548
12549 if (sz(digits) <= k) {
12550     ans = n;
12551 }
12552
12553     cout << ans - n << '\n';
12554 }
12555
12556 void output() {
12557 }
12558
12559 int main() {
12560     // freopen("parsadox2.txt","r+",stdin);
12561     // freopen("parsadox.txt","w+",stdout);
12562     ios::sync_with_stdio(false);
12563     cin.tie(0);
12564     cout.tie(0);
12565
12566     int number_of_testcases = 1;
12567     cin >> number_of_testcases;
12568     while (number_of_testcases--) {
12569         init();
12570
12571         input();
12572
12573         solve();
12574
12575         output();
12576     }
12577
12578     return 0;
12579 }
12580
12581 //SORTXOR
12582 #include <iostream>
12583 #include <string>
12584 #include <set>
12585 #include <map>
12586 #include <stack>
12587 #include <queue>
12588 #include <vector>
12589 #include <utility>
12590 #include <iomanip>
12591 #include <sstream>

```

```

12592 #include <bitset>
12593 #include <cstdlib>
12594 #include <iterator>
12595 #include <algorithm>
12596 #include <cstdio>
12597 #include <cctype>
12598 #include <cmath>
12599 #include <math.h>
12600 #include <ctime>
12601 #include <cstring>
12602 #include <unordered_set>
12603 #include <unordered_map>
12604 #include <cassert>
12605 #define int long long int
12606 #define pb push_back
12607 #define mp make_pair
12608 #define mod 1000000007
12609 #define vl vector<ll>
12610 #define all(c) (c).begin(), (c).end()
12611 using namespace std;
12612
12613 const int N=500023;
12614 bool vis[N];
12615 vector<int> adj[N];
12616 long long readInt(long long l, long long r, char endd){
12617     long long x=0;
12618     int cnt=0;
12619     int fi=-1;
12620     bool is_neg=false;
12621     while(true){
12622         char g=getchar();
12623         if(g=='-'){
12624             assert(fi==-1);
12625             is_neg=true;
12626             continue;
12627         }
12628         if('0'<=g && g<='9'){
12629             x*=10;
12630             x+=g-'0';
12631             if(cnt==0){
12632                 fi=g-'0';
12633             }
12634             cnt++;
12635             assert(fi!=0 || cnt==1);
12636             assert(fi!=0 || is_neg==false);
12637
12638             assert(!(cnt>19 || ( cnt==19 && fi>1) ));
12639         } else if(g==endd){
12640             if(is_neg){
12641                 x= -x;
12642             }
12643
12644             if(!(l <= x && x <= r))
12645             {
12646                 cerr << l << ' ' << r << ' ' << x << '\n';
12647                 assert(1 == 0);
12648             }
12649
12650             return x;
12651         } else {
12652             assert(false);
12653         }
12654     }
12655 }
12656 string readString(int l, int r, char endd){
12657     string ret="";
12658     int cnt=0;
12659     while(true){
12660         char g=getchar();

```

```

12661         assert(g!=-1);
12662         if(g==endd){
12663             break;
12664         }
12665         cnt++;
12666         ret+=g;
12667     }
12668     assert(l<=cnt && cnt<=r);
12669     return ret;
12670 }
12671 long long readIntSp(long long l,long long r){
12672     return readInt(l,r,' ');
12673 }
12674 long long readIntLn(long long l,long long r){
12675     return readInt(l,r,'\n');
12676 }
12677 string readStringLn(int l,int r){
12678     return readString(l,r,'\n');
12679 }
12680 string readStringSp(int l,int r){
12681     return readString(l,r,' ');
12682 }
12683
12684 int sumN = 0;
12685
12686 void solve()
12687 {
12688     int n = readInt(1, 1000, '\n');
12689     sumN += n;
12690     vector<int> a(n);
12691     set<int> s;
12692     for(int i = 0; i < n-1; i++) {
12693         a[i] = readInt(1, n, ' ');
12694         s.insert(a[i]);
12695     }
12696     a[n-1] = readInt(1, n, '\n');
12697     s.insert(a[n-1]);
12698     //assert(s.size() == n);
12699     cerr << "Input read successfully" <<endl;
12700     vector<int> b = a;
12701     sort(all(b));
12702
12703     map<int, vector<int>> indices;
12704     unordered_map<int, bool> vis;
12705     for(int i = 0; i < n; i++) {
12706         indices[b[i]].push_back(i);
12707     }
12708
12709     vector<vector<int> > cycles;
12710     for(int i = 0; i<n; i++){
12711         if(indices[a[i]].empty()) continue;
12712         vector<int> cycle;
12713         int j = i;
12714         while(!vis[j]){
12715             //cerr << j << " ";
12716             vis[j] = true;
12717             cycle.pb(j);
12718             int temp = j;
12719             j = indices[a[j]].back();
12720             indices[a[temp]].pop_back();
12721         }
12722         //cerr << endl;
12723         if(cycle.size() > 1)
12724             cycles.pb(cycle);
12725     }
12726
12727     int ans = 0;
12728     vector<pair<int, int>> index;
12729     vector<vector<int>> operations;

```

```

12730     for(auto cycle : cycles) {
12731         ans += cycle.size()+1;
12732         for(int i = 0; i < cycle.size(); i++) {
12733             index.pb({cycle[i], cycle.size()});
12734             operations.push_back(cycle);
12735             //cerr << cycle[i] << " " << cycle.size() << endl;
12736         }
12737         index.pb({cycle[0], cycle.size()});
12738         operations.push_back(cycle);
12739     }
12740
12741     //cerr << ans << endl;
12742
12743     for(int i = 0; i<ans; i++){
12744         int toUpdate = index[i].first;
12745         int updatedVal = 0;
12746         for(int j = 0; j < operations[i].size(); j++) {
12747             updatedVal ^= a[operations[i][j]];
12748         }
12749         a[toUpdate] = updatedVal;
12750     }
12751
12752     for(int i = 0; i < n; i++) {
12753         //cerr << a[i] << " ";
12754         assert(b[i] == a[i]);
12755     }
12756     //cerr << endl;
12757
12758     cout<<ans<<'\n';
12759     for(int i = 0; i < index.size(); i++) {
12760         cout<<index[i].first+1<<" "<<index[i].second<<'\n';
12761         for(int j = 0; j < operations[i].size(); j++) {
12762             cout<<operations[i][j]+1<<" ";
12763         }
12764         cout<<'\n';
12765     }
12766     cerr << "Operations printed" <<endl;
12767 }
12768
12769 int32_t main()
12770 {
12771     #ifndef ONLINE_JUDGE
12772     freopen("input.txt", "r", stdin);
12773     freopen("output.txt", "w", stdout);
12774     #endif
12775     ios_base::sync_with_stdio(false);
12776     cin.tie(NULL),cout.tie(NULL);
12777     int T=readInt(1,2000,'\n');
12778     cerr << "#Testcases read successfully" <<endl;
12779     while(T--){
12780         solve();
12781         //cout<<'\n';
12782     }
12783     cerr << sumN << '\n';
12784     assert(sumN <= 6000);
12785     cerr << "Time : " << 1000 * ((double)clock()) / (double)CLOCKS_PER_SEC << "ms\n";
12786 }
12787
12788 //CONCATPAL
12789 #ifdef WTSH
12790     #include <wtsh.h>
12791 #else
12792     #include <bits/stdc++.h>
12793     using namespace std;
12794     #define dbg(...)
12795 #endif
12796
12797 #define int long long
12798 #define endl "\n"

```

```

12799 #define sz(w) (int)(w.size())
12800 using pii = pair<int, int>;
12801
12802 const long long INF = 1e18;
12803
12804 const int N = 1e6 + 5;
12805
12806 // ----- Input Checker Start -----
12807
12808 long long readInt(long long l, long long r, char endd)
12809 {
12810     long long x = 0;
12811     int cnt = 0, fi = -1;
12812     bool is_neg = false;
12813     while(true)
12814     {
12815         char g = getchar();
12816         if(g == '-')
12817         {
12818             assert(fi == -1);
12819             is_neg = true;
12820             continue;
12821         }
12822         if('0' <= g && g <= '9')
12823         {
12824             x *= 10;
12825             x += g - '0';
12826             if(cnt == 0)
12827                 fi = g - '0';
12828             cnt++;
12829             assert(fi != 0 || cnt == 1);
12830             assert(fi != 0 || is_neg == false);
12831             assert(!(cnt > 19 || (cnt == 19 && fi > 1)));
12832         }
12833         else if(g == endd)
12834         {
12835             if(is_neg)
12836                 x = -x;
12837             if(!(l <= x && x <= r))
12838             {
12839                 cerr << "L: " << l << ", R: " << r << ", Value Found: " << x << '\n';
12840                 assert(false);
12841             }
12842             return x;
12843         }
12844         else
12845         {
12846             assert(false);
12847         }
12848     }
12849 }
12850
12851 string readString(int l, int r, char endd)
12852 {
12853     string ret = "";
12854     int cnt = 0;
12855     while(true)
12856     {
12857         char g = getchar();
12858         assert(g != -1);
12859         if(g == endd)
12860             break;
12861         cnt++;
12862         ret += g;
12863     }
12864     assert(l <= cnt && cnt <= r);
12865     return ret;
12866 }
12867

```

```

12868 long long readIntSp(long long l, long long r) { return readInt(l, r, ' '); }
12869 long long readIntLn(long long l, long long r) { return readInt(l, r, '\n'); }
12870 string readStringSp(int l, int r) { return readString(l, r, ' '); }
12871 string readStringLn(int l, int r) { return readString(l, r, '\n'); }
12872 void readEOF() { assert(getchar() == EOF); }
12873
12874 vector<int> readVectorInt(int n, long long l, long long r)
12875 {
12876     vector<int> a(n);
12877     for(int i = 0; i < n - 1; i++)
12878         a[i] = readIntSp(l, r);
12879     a[n - 1] = readIntLn(l, r);
12880     return a;
12881 }
12882
12883 // ----- Input Checker End -----
12884
12885 int sumN = 0;
12886 void solve()
12887 {
12888     int n = readIntSp(1, 2e5);
12889     int m = readIntLn(1, 2e5);
12890     sumN += n + m;
12891     string a = readStringLn(n, n);
12892     string b = readStringLn(m, m);
12893     for(auto &x: a) assert(x >= 'a' and x <= 'z');
12894     for(auto &x: b) assert(x >= 'a' and x <= 'z');
12895     if(n > m)
12896         swap(a, b), swap(n, m);
12897     array<int, 26> a_cnt{}, b_cnt{};
12898     for(auto &x: a)
12899         a_cnt[x - 'a']++;
12900     for(auto &x: b)
12901         b_cnt[x - 'a']++;
12902     bool ok = true;
12903     int odd = 0;
12904     for(int i = 0; i < 26; i++)
12905     {
12906         if(b_cnt[i] < a_cnt[i])
12907             ok = false;
12908         odd += (b_cnt[i] - a_cnt[i]) % 2;
12909     }
12910     if(odd <= 1 and ok)
12911         cout << "YES\n";
12912     else
12913         cout << "NO\n";
12914 }
12915
12916 int32_t main()
12917 {
12918     ios::sync_with_stdio(0);
12919     cin.tie(0);
12920     int T = readIntLn(1, 2e5);
12921     for(int tc = 1; tc <= T; tc++)
12922     {
12923         // cout << "Case #" << tc << ": ";
12924         solve();
12925     }
12926     assert(sumN <= 2e5);
12927     readEOF();
12928     return 0;
12929 }
12930
12931 //THREENUMBERS
12932 //Utkarsh.25dec
12933 #include <iostream>
12934 #include <cstdio>
12935 #include <cstdlib>

```

```

12937 #include <algorithm>
12938 #include <cmath>
12939 #include <vector>
12940 #include <set>
12941 #include <map>
12942 #include <unordered_set>
12943 #include <unordered_map>
12944 #include <queue>
12945 #include <ctime>
12946 #include <cassert>
12947 #include <complex>
12948 #include <string>
12949 #include <cstring>
12950 #include <chrono>
12951 #include <random>
12952 #include <bitset>
12953 #include <array>
12954 #define ll long long int
12955 #define pb push_back
12956 #define mp make_pair
12957 #define mod 1000000007
12958 #define vl vector<ll>
12959 #define all(c) (c).begin(), (c).end()
12960 using namespace std;
12961 ll power(ll a, ll b) {ll res=1; a%=mod; assert(b>=0); for(; b>=1; b--){if(b&1) res=res*a%mod; a
=a*a%mod;} return res;}
12962 ll modInverse(ll a){return power(a, mod-2);}
12963 const int N=500023;
12964 bool vis[N];
12965 vector<int> adj[N];
12966 long long readInt(long long l, long long r, char endd){
12967     long long x=0;
12968     int cnt=0;
12969     int fi=-1;
12970     bool is_neg=false;
12971     while(true){
12972         char g=getchar();
12973         if(g=='-'){
12974             assert(fi==-1);
12975             is_neg=true;
12976             continue;
12977         }
12978         if('0'<=g && g<='9'){
12979             x*=10;
12980             x+=g-'0';
12981             if(cnt==0){
12982                 fi=g-'0';
12983             }
12984             cnt++;
12985             assert(fi!=0 || cnt==1);
12986             assert(fi!=0 || is_neg==false);
12987
12988             assert(!(cnt>19 || ( cnt==19 && fi>1) ));
12989         } else if(g==endd){
12990             if(is_neg){
12991                 x=-x;
12992             }
12993
12994             if(!(l <= x && x <= r))
12995             {
12996                 cerr << l << ' ' << r << ' ' << x << '\n';
12997                 assert(1 == 0);
12998             }
12999
13000             return x;
13001         } else {
13002             assert(false);
13003         }
13004     }

```



```

13005 }
13006 string readString(int l,int r,char endd){
13007     string ret="";
13008     int cnt=0;
13009     while(true){
13010         char g=getchar();
13011         assert(g!=-1);
13012         if(g==endd){
13013             break;
13014         }
13015         cnt++;
13016         ret+=g;
13017     }
13018     assert(l<=cnt && cnt<=r);
13019     return ret;
13020 }
13021 long long readIntSp(long long l,long long r){
13022     return readInt(l,r,' ');
13023 }
13024 long long readIntLn(long long l,long long r){
13025     return readInt(l,r,'\n');
13026 }
13027 string readStringLn(int l,int r){
13028     return readString(l,r,'\n');
13029 }
13030 string readStringSp(int l,int r){
13031     return readString(l,r,' ');
13032 }
13033 void solve()
13034 {
13035     int A, B, C;
13036     A=readInt(1,1000000000,' ');
13037     B=readInt(1,1000000000,' ');
13038     C=readInt(1,1000000000,'\n');
13039     if((A%2 == B%2) && (B%2 == C%2))
13040     {
13041         int d=A+B+C-min({A,B,C});
13042         d/=2;
13043         cout<<(3*d-A-B-C)<<'\n';
13044     }
13045     else
13046         cout<<-1<<'\n';
13047 }
13048 int main()
13049 {
13050     #ifndef ONLINE_JUDGE
13051     freopen("input.txt", "r", stdin);
13052     freopen("output.txt", "w", stdout);
13053     #endif
13054     ios_base::sync_with_stdio(false);
13055     cin.tie(NULL),cout.tie(NULL);
13056     int T=readInt(1,10000,'\n');
13057     while(T--)
13058         solve();
13059     assert(getchar() == -1);
13060     cerr << "Time : " << 1000 * ((double)clock()) / (double)CLOCKS_PER_SEC << "ms\n";
13061 }
13062
13063 //CHRGES
13064 Setter's code (C++)
13065 #include <bits/stdc++.h>
13066 #define int long long int
13067 #define pb push_back
13068 #define mp make_pair
13069 #define mod 1000000007
13070 #define vl vector<ll>
13071 #define all(c) (c).begin(),(c).end()
13072 using namespace std;
13073

```

```

13074     const int N=500023;
13075     bool vis[N];
13076     vector<int> adj[N];
13077     long long readInt(long long l,long long r,char endd){
13078         long long x=0;
13079         int cnt=0;
13080         int fi=-1;
13081         bool is_neg=false;
13082         while(true){
13083             char g=getchar();
13084             if(g=='-'){
13085                 assert(fi==-1);
13086                 is_neg=true;
13087                 continue;
13088             }
13089             if('0'<=g && g<='9'){
13090                 x*=10;
13091                 x+=g-'0';
13092                 if(cnt==0){
13093                     fi=g-'0';
13094                 }
13095                 cnt++;
13096                 assert(fi!=0 || cnt==1);
13097                 assert(fi!=0 || is_neg==false);

13098                 assert(!(cnt>19 || ( cnt==19 && fi>1) ));
13099             } else if(g==endd){
13100                 if(is_neg){
13101                     x= -x;
13102                 }

13103                 if(!(l <= x && x <= r))
13104                 {
13105                     cerr << l << ' ' << r << ' ' << x << '\n';
13106                     assert(1 == 0);
13107                 }

13108                 return x;
13109             } else {
13110                 assert(false);
13111             }
13112         }
13113     }
13114 }
13115
13116 string readString(int l,int r,char endd){
13117     string ret="";
13118     int cnt=0;
13119     while(true){
13120         char g=getchar();
13121         assert(g!=-1);
13122         if(g==endd){
13123             break;
13124         }
13125         cnt++;
13126         ret+=g;
13127     }
13128     assert(l<=cnt && cnt<=r);
13129     return ret;
13130 }
13131
13132 long long readIntSp(long long l,long long r){
13133     return readInt(l,r,' ');
13134 }
13135
13136 long long readIntLn(long long l,long long r){
13137     return readInt(l,r,'\n');
13138 }
13139
13140 string readStringLn(int l,int r){
13141     return readString(l,r,'\n');
13142 }
13143
13144 string readStringSp(int l,int r){
13145     return readString(l,r,' ');
13146 }

```

```

13143 }
13144
13145 int sumN = 0;
13146
13147 void solve()
13148 {
13149     int n = readInt(1, 100000, '\n');
13150     sumN += n;
13151     string s = readStringLn(1, n);
13152     assert(s.size() == n);
13153     for(int i = 0; i < n; i++) {
13154         assert(s[i] == '0' || s[i] == '+' || s[i] == '-');
13155     }
13156
13157     int left[n], right[n];
13158     bool lPos[n], rPos[n];
13159     int last = -1;
13160     bool pos = true;
13161     for(int i = 0; i < n; i++){
13162         if(s[i] == '+'){
13163             pos = true;
13164             last = i;
13165         }
13166         else if(s[i] == '-'){
13167             pos = false;
13168             last = i;
13169         }
13170         else{
13171             if(last == -1){
13172                 left[i] = INT_MAX;
13173             }
13174             else{
13175                 left[i] = i - last;
13176                 lPos[i] = pos;
13177             }
13178         }
13179     }
13180     last = -1;
13181     pos = true;
13182     for(int i = n-1; i >= 0; i--){
13183         if(s[i] == '+'){
13184             pos = true;
13185             last = i;
13186         }
13187         else if(s[i] == '-'){
13188             pos = false;
13189             last = i;
13190         }
13191         else{
13192             if(last == -1){
13193                 right[i] = INT_MAX;
13194             }
13195             else{
13196                 right[i] = last - i;
13197                 rPos[i] = pos;
13198             }
13199         }
13200     }
13201
13202     int ans = 0;
13203     for(int i = 0; i < n; i++){
13204         if(s[i] == '0'){
13205             if((left[i] == INT_MAX) && (right[i] == INT_MAX)){
13206                 ans ++;
13207             }
13208             else if((left[i] != INT_MAX) && (right[i] != INT_MAX) && (left[i] == right[i])){
13209                 if(lPos[i] != rPos[i]){
13210                     ans ++;

```

```

13211         }
13212     }
13213 }
13214 }
13215 cout << ans << '\n';
13216 }
13217
13218 int32_t main()
13219 {
13220     #ifndef ONLINE_JUDGE
13221     freopen("input.txt", "r", stdin);
13222     freopen("output.txt", "w", stdout);
13223     #endif
13224     ios_base::sync_with_stdio(false);
13225     cin.tie(NULL), cout.tie(NULL);
13226     int T=readInt(1,2000,'\n');
13227     while(T--){
13228         solve();
13229     }
13230     assert(getchar()==-1);
13231     cerr << sumN << '\n';
13232     assert(sumN <= 200000);
13233     cerr << "Time : " << 1000 * ((double)clock()) / (double)CLOCKS_PER_SEC << "ms\n";
13234 }
13235
13236 //KPAL
13237 #ifdef WTSH
13238     #include <wtsh.h>
13239 #else
13240     #include <bits/stdc++.h>
13241     using namespace std;
13242     #define dbg(...)
13243 #endif
13244
13245 #define int long long
13246 #define endl "\n"
13247 #define sz(w) (int)(w.size())
13248 using pii = pair<int, int>;
13249
13250 const long long INF = 1e18;
13251
13252 const int N = 1e6 + 5;
13253
13254 // ----- Input Checker Start -----
13255
13256 long long readInt(long long l, long long r, char endd)
13257 {
13258     long long x = 0;
13259     int cnt = 0, fi = -1;
13260     bool is_neg = false;
13261     while(true)
13262     {
13263         char g = getchar();
13264         if(g == '-')
13265         {
13266             assert(fi == -1);
13267             is_neg = true;
13268             continue;
13269         }
13270         if('0' <= g && g <= '9')
13271         {
13272             x *= 10;
13273             x += g - '0';
13274             if(cnt == 0)
13275                 fi = g - '0';
13276             cnt++;
13277             assert(fi != 0 || cnt == 1);
13278             assert(fi != 0 || is_neg == false);
13279             assert(!(cnt > 19 || (cnt == 19 && fi > 1)));

```

```

13280     }
13281     else if(g == endd)
13282     {
13283         if(is_neg)
13284             x = -x;
13285         if(!(l <= x && x <= r))
13286         {
13287             cerr << "L: " << l << ", R: " << r << ", Value Found: " << x << '\n';
13288             assert(false);
13289         }
13290         return x;
13291     }
13292     else
13293     {
13294         assert(false);
13295     }
13296 }
13297 }
13298
13299 string readString(int l, int r, char endd)
13300 {
13301     string ret = "";
13302     int cnt = 0;
13303     while(true)
13304     {
13305         char g = getchar();
13306         assert(g != -1);
13307         if(g == endd)
13308             break;
13309         cnt++;
13310         ret += g;
13311     }
13312     assert(l <= cnt && cnt <= r);
13313     return ret;
13314 }
13315
13316 long long readIntSp(long long l, long long r) { return readInt(l, r, ' '); }
13317 long long readIntLn(long long l, long long r) { return readInt(l, r, '\n'); }
13318 string readStringSp(int l, int r) { return readString(l, r, ' '); }
13319 string readStringLn(int l, int r) { return readString(l, r, '\n'); }
13320 void readEOF() { assert(getchar() == EOF); }
13321
13322 vector<int> readVectorInt(int n, long long l, long long r)
13323 {
13324     vector<int> a(n);
13325     for(int i = 0; i < n - 1; i++)
13326         a[i] = readIntSp(l, r);
13327     a[n - 1] = readIntLn(l, r);
13328     return a;
13329 }
13330
13331 // ----- Input Checker End -----
13332
13333 int sumN = 0;
13334
13335 void solve()
13336 {
13337     int n = readIntSp(1, 1e5);
13338     int k = readIntLn(1, n);
13339     sumN += n;
13340     vector<int> a = readVectorInt(n, 1, 1e6);
13341     vector<int> b = a;
13342     reverse(b.begin(), b.end());
13343     if(a == b)
13344         cout << "YES\n";
13345     else
13346     {
13347         if(k == n)
13348             cout << "NO\n";

```

```

13349         else if(n % 2 == 1 or k % 2 == 1)
13350             cout << "YES\n";
13351         else
13352         {
13353             int sum = accumulate(a.begin(), a.end(), 0LL);
13354             if(sum % 2 == 0)
13355                 cout << "YES\n";
13356             else
13357                 cout << "NO\n";
13358         }
13359     }
13360 }
13361
13362 int32_t main()
13363 {
13364     ios::sync_with_stdio(0);
13365     cin.tie(0);
13366     int T = readIntLn(1, 1e5);
13367     for(int tc = 1; tc <= T; tc++)
13368     {
13369         // cout << "Case #" << tc << ": ";
13370         solve();
13371     }
13372     assert(sumN <= 2e5);
13373     readEOF();
13374     return 0;
13375 }
13376
13377 //KFOREST
13378 #include <bits/stdc++.h>
13379 using namespace std;
13380 #define ll long long
13381 #define ull unsigned long long
13382 #define pb(e) push_back(e)
13383 #define sv(a) sort(a.begin(), a.end())
13384 #define sa(a,n) sort(a,a+n)
13385 #define mp(a,b) make_pair(a,b)
13386 #define vf first
13387 #define vs second
13388 #define ar array
13389 #define all(x) x.begin(), x.end()
13390 const int inf = 0x3f3f3f3f;
13391 const int mod = 1000000007;
13392 const double PI=3.14159265358979323846264338327950288419716939937510582097494459230;
13393
13394 mt19937_64 RNG(chrono::steady_clock::now().time_since_epoch().count());
13395
13396 bool remender(ll a , ll b){return a%b;}
13397
13398 //freopen("problemname.in", "r", stdin);
13399 //freopen("problemname.out", "w", stdout);
13400
13401 const int N = 200003;
13402
13403 vector<int> adj[N];
13404 int arr[N];
13405 int cmp;
13406
13407 int dfs(int node , int par , int desire){
13408     int cur = arr[node];
13409     for(int i : adj[node]){
13410         if(i == par)continue;
13411         cur ^= dfs(i , node , desire);
13412     }
13413     if((cur&desire)==desire){
13414         cmp++;
13415         cur = 0;
13416     }
13417     return cur;

```

```

13418 }
13419
13420 int solve(int k , int n){
13421     int ans = 0;
13422     for(int i = 30; i >= 0; i--){
13423         ans += (1 << i);
13424         cmp = 0;
13425         int x = dfs(1,1,ans);
13426         if((x&ans) != 0) cmp = 0;
13427         if(cmp < k || (cmp - k) % 2 == 1){
13428             ans -= (1<<i);
13429         }
13430     }
13431     return ans;
13432 }
13433
13434 int main(){
13435     ios_base::sync_with_stdio(false);
13436     cin.tie(NULL);
13437     int t;cin >> t;while(t--){
13438         int n;
13439         cin >> n;
13440         int k;
13441         cin >> k;
13442         for(int i = 1; i <= n; i++)cin >> arr[i];
13443         for(int i = 0; i < n - 1; i++){
13444             int u , v;
13445             cin >> u >> v;
13446             adj[u].pb(v);
13447             adj[v].pb(u);
13448         }
13449         cout << solve(k , n) << '\n';
13450         for(int i = 0; i <= n; i++)adj[i].clear();
13451     }
13452     return 0;
13453 }
13454
13455 //OPERATION2
13456 #include <map>
13457 #include <set>
13458 #include <cmath>
13459 #include <ctime>
13460 #include <queue>
13461 #include <stack>
13462 #include <cstdio>
13463 #include <cstdlib>
13464 #include <vector>
13465 #include <cstring>
13466 #include <algorithm>
13467 #include <iostream>
13468 using namespace std;
13469 typedef double db;
13470 typedef long long ll;
13471 typedef unsigned long long ull;
13472 const int N=1000010;
13473 const int LOGN=28;
13474 const ll TMD=0;
13475 const ll INF=2147483647;
13476 int T,n,ans;
13477 int a[N],lg2[N];
13478 int OR[N][LOGN],AND[N][LOGN];
13479
13480 void init()
13481 {
13482     for(int i=1;i<=n;i++) a[i+n]=a[i];
13483     n<<=1;
13484     for(int i=1;i<=n;i++) lg2[i]=(int)log2(i);
13485     for(int i=1;i<=n;i++) OR[i][0]=AND[i][0]=a[i];
13486     for(int i=1;i<LOGN;i++)

```

```

13487     {
13488         for(int j=1;j<=n;j++)
13489         {
13490             int p=j+(1<<(i-1));
13491             if(p>n) OR[j][i]=OR[j][i-1],AND[j][i]=AND[j][i-1];
13492             else OR[j][i]=(OR[j][i-1]|OR[p][i-1]),AND[j][i]=(AND[j][i-1]&AND[p][i-1]);
13493         }
13494     }
13495     n>>=1;
13496 }
13497
13498 int getOR(int L,int R)
13499 {
13500     int t=lg2[R-L+1];
13501     return (OR[L][t]|OR[R-(1<<t)+1][t]);
13502 }
13503
13504 int getAND(int L,int R)
13505 {
13506     int t=lg2[R-L+1];
13507     return (AND[L][t]&AND[R-(1<<t)+1][t]);
13508 }
13509
13510 int main()
13511 {
13512     scanf("%d",&T);
13513     while(T--)
13514     {
13515         scanf("%d",&n);
13516         for(int i=1;i<=n;i++) scanf("%d",&a[i]);
13517         init();
13518         ans=0;
13519         for(int i=1;i<=n;i++)
13520         {
13521             int cur=i,L,R,M;
13522             while(1)
13523             {
13524                 ans=max(ans,(int)abs(getOR(i,cur)-getAND(cur+1,i+n-1)));
13525                 L=cur;R=i+n-1;
13526                 while(L+1!=R)
13527                 {
13528                     M=(L+R)>>1;
13529                     if(getOR(i,cur)==getOR(i,M)) L=M;
13530                     else R=M;
13531                 }
13532                 if(R==i+n-1) break;
13533                 cur=R;
13534             }
13535         }
13536         printf("%d\n",ans);
13537     }
13538
13539     return 0;
13540 }
13541
13542 //EXPVALUE
13543 //#pragma GCC target ("avx2")
13544 #pragma GCC optimize ("O3")
13545 #pragma GCC optimize ("unroll-loops")
13546
13547
13548 #include <bits/stdc++.h>
13549 #include <ext/pb_ds/tree_policy.hpp>
13550 #include <ext/pb_ds/assoc_container.hpp>
13551 using namespace __gnu_pbds;
13552 using namespace std;
13553 #define ll long long
13554 const ll INF_MUL=1e13;
13555 const ll INF_ADD=1e18;

```



```

13556 #define pb push_back
13557 #define mp make_pair
13558 #define nline "\n"
13559 #define f first
13560 #define s second
13561 #define pll pair<ll,ll>
13562 #define all(x) x.begin(),x.end()
13563 #define vl vector<ll>
13564 #define vvl vector<vector<ll>>
13565 #define vvvl vector<vector<vector<ll>>>
13566 #ifndef ONLINE_JUDGE
13567 #define debug(x) cerr<<"x<<" "; _print(x); cerr<<nline;
13568 #else
13569 #define debug(x);
13570 #endif
13571 void _print(ll x){cerr<<x;}
13572 void _print(char x){cerr<<x;}
13573 void _print(string x){cerr<<x;}
13574 mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());
13575 template<class T,class V> void _print(pair<T,V> p) {cerr<<"{"; _print(p.first);cerr<<"",
; _print(p.second);cerr<<""}";}
13576 template<class T>void _print(vector<T> v) {cerr<<" [ "; for (T i:v){_print(i);cerr<<" "
; }cerr<<""]";}
13577 template<class T>void _print(set<T> v) {cerr<<" [ "; for (T i:v){_print(i); cerr<<" ";}
cerr<<""]";}
13578 template<class T>void _print(multiset<T> v) {cerr<<" [ "; for (T i:v){_print(i);cerr<<
" ";}cerr<<""]";}
13579 template<class T,class V>void _print(map<T, V> v) {cerr<<" [ "; for(auto i:v) {_print(i
);cerr<<" ";} cerr<<""]";}
13580 typedef tree<ll, null_type, less<ll>, rb_tree_tag, tree_order_statistics_node_update>
ordered_set;
13581 typedef tree<ll, null_type, less_equal<ll>, rb_tree_tag,
tree_order_statistics_node_update> ordered_multiset;
13582 typedef tree<pair<ll,ll>, null_type, less<pair<ll,ll>>, rb_tree_tag,
tree_order_statistics_node_update> ordered_pset;
13583 //-----
-----

13584 const ll MOD=998244353;
13585 const ll MAX=200200;
13586 ll binpow(ll a,ll b,ll MOD){
13587     ll ans=1;
13588     a%=MOD;
13589     while(b){
13590         if(b&1)
13591             ans=(ans*a)%MOD;
13592         b/=2;
13593         a=(a*a)%MOD;
13594     }
13595     return ans;
13596 }
13597 ll inverse(ll a,ll MOD){
13598     return binpow(a,MOD-2,MOD);
13599 }
13600 ll getv(ll p,ll n){
13601     ll num=binpow(p,n,MOD)-1+MOD;
13602     ll den=inverse(p+MOD-1,MOD);
13603     num=(num*den)%MOD;
13604     return num;
13605 }
13606 void solve(){
13607     ll n,p; cin>>n>>p;
13608     p=inverse(p,MOD);
13609     for(ll i=1;i<=n;i++){
13610         ll ans=(getv(p,i)*getv(p,i)+getv(p*p,i))%MOD;
13611         ans=(ans*inverse(4,MOD))%MOD;
13612         cout<<ans<<" \n"[i==n];
13613     }
13614     return;

```

```

13615 }
13616 int main()

13617 {
13618     ios_base::sync_with_stdio(false);
13619     cin.tie(NULL);
13620     #ifndef ONLINE_JUDGE
13621     freopen("input.txt", "r", stdin);
13622     freopen("output.txt", "w", stdout);
13623     freopen("error.txt", "w", stderr);
13624     #endif
13625     ll test_cases=1;
13626     //cin>>test_cases;
13627     while(test_cases--){
13628         solve();
13629     }
13630     cout<<fixed<<setprecision(10);
13631     cerr<<"Time:"<<1000*((double)clock())/((double)CLOCKS_PER_SEC)<<"ms\n";
13632 }
13633
13634
13635
13636 //NOTDIVISIBLE
13637 #include <iostream>
13638 using namespace std;
13639
13640 int main() {
13641     int t; cin>>t;
13642     while(t--){
13643         int n; cin>>n; n++;
13644         while(--n){
13645             cout<<(n&1)<<" ";
13646         }
13647         cout<<"\n";
13648     }
13649     return 0;
13650 }
13651
13652 //CONSTRRAY
13653 // #pragma GCC target ("avx2")
13654 #pragma GCC optimize ("O3")
13655 #pragma GCC optimize ("unroll-loops")
13656
13657
13658 #include <bits/stdc++.h>
13659 #include <ext/pb_ds/tree_policy.hpp>
13660 #include <ext/pb_ds/assoc_container.hpp>
13661 using namespace __gnu_pbds;
13662 using namespace std;
13663 #define ll long long
13664 const ll INF_MUL=1e13;
13665 const ll INF_ADD=1e18;
13666 #define pb push_back
13667 #define mp make_pair
13668 #define nline "\n"
13669 #define f first
13670 #define s second
13671 #define pll pair<ll,ll>
13672 #define all(x) x.begin(),x.end()
13673 #define vl vector<ll>
13674 #define vvl vector<vector<ll>>
13675 #define vvvl vector<vector<vector<ll>>>
13676 #ifndef ONLINE_JUDGE
13677 #define debug(x) cerr<<#x<<" "; _print(x); cerr<<nline;
13678 #else
13679 #define debug(x);
13680 #endif
13681 void _print(ll x){cerr<<x;}

```

```

13682 void _print(char x){cerr<<x;}
13683 void _print(string x){cerr<<x;}
13684 mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());
13685 template<class T,class V> void _print(pair<T,V> p) {cerr<<"{"; _print(p.first);cerr<<"",
; _print(p.second);cerr<<"}";}
13686 template<class T>void _print(vector<T> v) {cerr<<" [ "; for (T i:v){_print(i);cerr<<" "
;};cerr<<""]";}
13687 template<class T>void _print(set<T> v) {cerr<<" [ "; for (T i:v){_print(i); cerr<<" "};
cerr<<""]";}
13688 template<class T>void _print(multiset<T> v) {cerr<<" [ "; for (T i:v){_print(i);cerr<<
" "};cerr<<""]";}
13689 template<class T,class V>void _print(map<T, V> v) {cerr<<" [ "; for(auto i:v) {_print(i
);cerr<<" "}; cerr<<""]";}
13690 typedef tree<ll, null_type, less<ll>, rb_tree_tag, tree_order_statistics_node_update>
ordered_set;
13691 typedef tree<ll, null_type, less_equal<ll>, rb_tree_tag,
tree_order_statistics_node_update> ordered_multiset;
13692 typedef tree<pair<ll,ll>, null_type, less<pair<ll,ll>>, rb_tree_tag,
tree_order_statistics_node_update> ordered_pset;
13693 //-----
-----
13694 const ll MOD=998244353;
13695 const ll MAX=500500;
13696 void solve(){
13697     ll n; cin>>n;
13698     if(n&1){
13699         cout<<"-1\n";
13700     }
13701     else{
13702         if(n==4){
13703             cout<<"0 5 343 -100\n";
13704             return;
13705         }
13706         cout<<"1";
13707         ll use=-2;
13708         for(ll i=2;i<n;i++){
13709             cout<<" "<<use;
13710             use*=-1;
13711         }
13712         cout<<" -1\n";
13713     }
13714     return;
13715 }
13716 int main()

13717 {
13718     ios_base::sync_with_stdio(false);
13719     cin.tie(NULL);
13720     #ifndef ONLINE_JUDGE
13721     freopen("input.txt", "r", stdin);
13722     freopen("output.txt", "w", stdout);
13723     freopen("error.txt", "w", stderr);
13724     #endif
13725     ll test_cases=1;
13726     cin>>test_cases;
13727     while(test_cases--){
13728         solve();
13729     }
13730     cout<<fixed<<setprecision(10);
13731     cerr<<"Time:"<<1000*((double)clock())/((double)CLOCKS_PER_SEC<<"ms\n";
13732 }
13733
13734 //COUNTISFUN
13735 // #pragma GCC target ("avx2")
13736 #pragma GCC optimize ("O3")
13737 #pragma GCC optimize ("unroll-loops")
13738

```

```

13739
13740 #include <bits/stdc++.h>
13741 #include <ext/pb_ds/tree_policy.hpp>
13742 #include <ext/pb_ds/assoc_container.hpp>
13743 using namespace __gnu_pbds;
13744 using namespace std;
13745 #define ll long long
13746 const ll INF_MUL=1e13;
13747 const ll INF_ADD=1e18;
13748 #define pb push_back
13749 #define mp make_pair
13750 #define nline "\n"
13751 #define f first
13752 #define s second
13753 #define pll pair<ll,ll>
13754 #define all(x) x.begin(),x.end()
13755 #define vl vector<ll>
13756 #define vvl vector<vector<ll>>
13757 #define vvvl vector<vector<vector<ll>>>
13758 #ifndef ONLINE_JUDGE
13759 #define debug(x) cerr<<#x<<" "; _print(x); cerr<<nline;
13760 #else
13761 #define debug(x);
13762 #endif
13763 void _print(ll x){cerr<<x;}
13764 void _print(char x){cerr<<x;}
13765 void _print(string x){cerr<<x;}
13766 mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());
13767 template<class T,class V> void _print(pair<T,V> p) {cerr<<"{"; _print(p.first);cerr<<" ";
; _print(p.second);cerr<<"}";}
13768 template<class T>void _print(vector<T> v) {cerr<<" [ "; for (T i:v){_print(i);cerr<<" "
; }cerr<<" ]";}
13769 template<class T>void _print(set<T> v) {cerr<<" [ "; for (T i:v){_print(i); cerr<<" ";}
cerr<<" ]";}
13770 template<class T>void _print(multiset<T> v) {cerr<<" [ "; for (T i:v){_print(i);cerr<<
" ";}cerr<<" ]";}
13771 template<class T,class V>void _print(map<T, V> v) {cerr<<" [ "; for(auto i:v) {_print(i
);cerr<<" ";} cerr<<" ]";}
13772 typedef tree<ll, null_type, less<ll>, rb_tree_tag, tree_order_statistics_node_update>
ordered_set;
13773 typedef tree<ll, null_type, less_equal<ll>, rb_tree_tag,
tree_order_statistics_node_update> ordered_multiset;
13774 typedef tree<pair<ll,ll>, null_type, less<pair<ll,ll>>, rb_tree_tag,
tree_order_statistics_node_update> ordered_pset;
13775 //-----
-----

13776 const ll MOD=998244353;
13777 const ll MAX=500500;
13778 vector<ll> fact(MAX+2,1),inv_fact(MAX+2,1);
13779 ll binpow(ll a,ll b,ll MOD){
13780     ll ans=1;
13781     a%=MOD;
13782     while(b){
13783         if(b&1)
13784             ans=(ans*a)%MOD;
13785         b/=2;
13786         a=(a*a)%MOD;
13787     }
13788     return ans;
13789 }
13790 ll inverse(ll a,ll MOD){
13791     return binpow(a,MOD-2,MOD);
13792 }
13793 void precompute(ll MOD){
13794     for(ll i=2;i<MAX;i++){
13795         fact[i]=(fact[i-1]*i)%MOD;
13796     }
13797     inv_fact[MAX-1]=inverse(fact[MAX-1],MOD);

```

```

13798     for(ll i=MAX-2;i>=0;i--){
13799         inv_fact[i]=(inv_fact[i+1]*(i+1))%MOD;
13800     }
13801 }
13802 ll nCr(ll a,ll b,ll MOD){
13803     if((a<0)|| (a<b)|| (b<0))
13804         return 0;
13805     ll denom=(inv_fact[b]*inv_fact[a-b])%MOD;
13806     return (denom*fact[a])%MOD;
13807 }
13808 vector<ll> power(MAX,1);
13809 void solve(){
13810     ll n; cin>>n;
13811     ll ans=(n*power[2*n])%MOD;
13812     for(ll l=0;l<=n;l++){
13813         for(ll r=0;l+r<=n;r++){
13814             ll now=min(l,r)+1;
13815             now/=2;
13816             ll ways=nCr(n,l,MOD)*nCr(n-l,r,MOD);
13817             ways%=MOD;
13818             ways=(ways*power[n-l-r])%MOD;
13819             ans=(ans-now*ways)%MOD;
13820         }
13821     }
13822     ans=(ans+MOD)%MOD;
13823     cout<<ans<<endl;
13824     return;
13825 }
13826 int main()

13827 {
13828     ios_base::sync_with_stdio(false);
13829     cin.tie(NULL);
13830     #ifndef ONLINE_JUDGE
13831     freopen("input.txt", "r", stdin);
13832     freopen("output.txt", "w", stdout);
13833     freopen("error.txt", "w", stderr);
13834     #endif
13835     ll test_cases=1;
13836     cin>>test_cases;
13837     precompute(MOD);
13838     for(ll i=1;i<MAX;i++){
13839         power[i]=(power[i-1]*2)%MOD;
13840     }
13841     while(test_cases--){
13842         solve();
13843     }
13844     cout<<fixed<<setprecision(10);
13845     cerr<<"Time:"<<1000*((double)clock())/(double)CLOCKS_PER_SEC<<"ms\n";
13846 }

13847 //MAKEARRYODD
13848 #include <bits/stdc++.h>
13849 #include <ext/pb_ds/assoc_container.hpp>
13850 #include <ext/pb_ds/tree_policy.hpp>
13851 #define IOS std::ios::sync_with_stdio(false); cin.tie(NULL);cout.tie(NULL);
13852 #define ll long long
13853 using namespace std;
13854 using namespace __gnu_pbds;
13855 ll int mod=1e9+7;//998244353;
13856 typedef tree<pair<int,string>, null_type, less<int>, rb_tree_tag,
13857 tree_order_statistics_node_update> ordered_set;
13858 #define PI 3.14159265
13859
13860
13861 ll int mul(ll int x, ll int y)
13862 {
13863     return (x * 111 * y) % mod;

```

```

13864     }
13865
13866
13867 ll int add(ll int x,ll int y)
13868 {
13869     x += y;
13870     while(x >= mod) x -= mod;
13871     while(x < 0) x += mod;
13872     return x;
13873 }
13874
13875
13876 long long power(long long a, long long b,ll m) {
13877     a %= m;
13878     long long res = 1;
13879     while (b > 0) {
13880         if (b & 1)
13881             res = (res*a)%m;
13882         a =(a*a)%m;
13883         b >>= 1;
13884     }
13885     return res%m;
13886 }
13887
13888
13889
13890 int main() {
13891     IOS;
13892     ll int t;
13893     cin>>t;
13894     while(t--)
13895     {
13896         ll int n,x;
13897         cin>>n>>x;
13898         ll int a[n];
13899         ll int cnt=0;
13900         ll int ans=-1;
13901         for(int i=0;i<n;i++)
13902         {
13903             cin>>a[i];
13904             if(a[i]%2==0)
13905                 cnt++;
13906         }
13907         if(x%2)
13908         {
13909             ans=(cnt+1)/2;
13910         }
13911         else {
13912             if(cnt<n)
13913                 ans=cnt;
13914         }
13915         cout<<ans<<endl;
13916     }
13917 }
13918
13919 //EQUALHAMMING
13920 //Utkarsh.25dec
13921 #include <iostream>
13922 #include <cstdio>
13923 #include <cstdlib>
13924 #include <algorithm>
13925 #include <cmath>
13926 #include <vector>
13927 #include <set>
13928 #include <map>
13929 #include <unordered_set>
13930 #include <unordered_map>
13931 #include <queue>
13932 #include <ctime>

```

```

13933 #include <cassert>
13934 #include <complex>
13935 #include <string>
13936 #include <cstring>
13937 #include <chrono>
13938 #include <random>
13939 #include <bitset>
13940 #include <array>
13941 #define ll long long int
13942 #define pb push_back
13943 #define mp make_pair
13944 #define mod 1000000007
13945 #define vl vector<ll>
13946 #define all(c) (c).begin(), (c).end()
13947 using namespace std;
13948 ll power(ll a, ll b) {ll res=1; a%=mod; assert(b>=0); for(;b;b>=1){if(b&1) res=res*a%mod; a
=a*a%mod;} return res;}
13949 ll modInverse(ll a){return power(a, mod-2);}
13950 const int N=500023;
13951 bool vis[N];
13952 vector<int> adj[N];
13953 long long readInt(long long l, long long r, char endd){
13954     long long x=0;
13955     int cnt=0;
13956     int fi=-1;
13957     bool is_neg=false;
13958     while(true){
13959         char g=getchar();
13960         if(g=='-'){
13961             assert(fi==-1);
13962             is_neg=true;
13963             continue;
13964         }
13965         if('0'<=g && g<='9'){
13966             x*=10;
13967             x+=g-'0';
13968             if(cnt==0){
13969                 fi=g-'0';
13970             }
13971             cnt++;
13972             assert(fi!=0 || cnt==1);
13973             assert(fi!=0 || is_neg==false);
13974
13975             assert(!(cnt>19 || (cnt==19 && fi>1)));
13976         } else if(g==endd){
13977             if(is_neg){
13978                 x=-x;
13979             }
13980
13981             if(!(l <= x && x <= r))
13982             {
13983                 cerr << l << ' ' << r << ' ' << x << '\n';
13984                 assert(1==0);
13985             }
13986
13987             return x;
13988         } else {
13989             assert(false);
13990         }
13991     }
13992 }
13993 string readString(int l, int r, char endd){
13994     string ret="";
13995     int cnt=0;
13996     while(true){
13997         char g=getchar();
13998         assert(g!=-1);
13999         if(g==endd){
14000             break;

```

```

14001         }
14002         cnt++;
14003         ret+=g;
14004     }
14005     assert(l<=cnt && cnt<=r);
14006     return ret;
14007 }
14008 long long readIntSp(long long l,long long r){
14009     return readInt(l,r,' ');
14010 }
14011 long long readIntLn(long long l,long long r){
14012     return readInt(l,r,'\n');
14013 }
14014 string readStringLn(int l,int r){
14015     return readString(l,r,'\n');
14016 }
14017 string readStringSp(int l,int r){
14018     return readString(l,r,' ');
14019 }
14020 int sumN=0;
14021 ll fact[N];
14022 ll invfact[N];
14023 ll inv[N];
14024 void factorialsComputation()
14025 {
14026     inv[0]=inv[1]=1;
14027     fact[0]=fact[1]=1;
14028     invfact[0]=invfact[1]=1;
14029     for(int i=2;i<N;i++)
14030     {
14031         inv[i]=(inv[mod%i]*(mod-mod/i))%mod;
14032         fact[i]=(fact[i-1]*i)%mod;
14033         invfact[i]=(invfact[i-1]*inv[i])%mod;
14034     }
14035 }
14036 ll ncr(ll n,ll r)
14037 {
14038     ll ans=fact[n]*invfact[r];
14039     ans%=mod;
14040     ans*=invfact[n-r];
14041     ans%=mod;
14042     return ans;
14043 }
14044 void solve()
14045 {
14046     int n=readInt(1,200000,'\n');
14047     sumN+=n;
14048     assert(sumN<=200000);
14049     string A=readString(n,n,'\n');
14050     string B=readString(n,n,'\n');
14051     int good=0,bad=0;
14052     for(int i=0;i<n;i++)
14053     {
14054         assert(A[i]=='0' || A[i]=='1');
14055         assert(B[i]=='0' || B[i]=='1');
14056         if(A[i]==B[i])
14057             good++;
14058         else
14059             bad++;
14060     }
14061     if(bad%2==1)
14062     {
14063         cout<<0<<'\n';
14064         return;
14065     }
14066     ll ans=power(2,good)*ncr(bad,bad/2);
14067     ans%=mod;
14068     cout<<ans<<'\n';
14069 }

```



```

14070 int main()
14071 {
14072     #ifndef ONLINE_JUDGE
14073     freopen("input.txt", "r", stdin);
14074     freopen("output.txt", "w", stdout);
14075     #endif
14076     ios_base::sync_with_stdio(false);
14077     cin.tie(NULL), cout.tie(NULL);
14078     factorialsComputation();
14079     int T=readInt(1,1000,'\n');
14080     while(T--)
14081         solve();
14082     assert(getchar()==-1);
14083     cerr << "Time : " << 1000 * ((double)clock()) / (double)CLOCKS_PER_SEC << "ms\n";
14084 }
14085
14086 //CHEFFFUNC
14087 #include <bits/stdc++.h>
14088 using namespace std;
14089 #define int long long
14090 #define pb push_back
14091 #define ppb pop_back
14092 #define pf push_front
14093 #define ppf pop_front
14094 #define all(x) (x).begin(), (x).end()
14095 #define uniq(v) (v).erase(unique(all(v)), (v).end())
14096 #define sz(x) (int)((x).size())
14097 #define fr first
14098 #define sc second
14099 #define pii pair<int,int>
14100 #define rep(i,a,b) for(int i=a;i<b;i++)
14101 #define mem1(a) memset(a,-1,sizeof(a))
14102 #define mem0(a) memset(a,0,sizeof(a))
14103 #define ppc __builtin_popcount
14104 #define ppcll __builtin_popcountll
14105 #define debug(x) cout<<(x)<<'\n';
14106
14107
14108 template<typename T1,typename T2>istream& operator>>(istream& in,pair<T1,T2> &a){in>>a.
fr>>a.sc;return in;}
14109 template<typename T1,typename T2>ostream& operator<<(ostream& out,pair<T1,T2> a){out<<a.
fr<<" "<<a.sc;return out;}
14110 template<typename T,typename T1>T amax(T &a,T1 b){if(b>a)a=b;return a;}
14111 template<typename T,typename T1>T amin(T &a,T1 b){if(b<a)a=b;return a;}
14112
14113 const long long INF=1e18;
14114 const int32_t M=1e9+7;
14115 const int32_t MM=998244353;
14116
14117 const int N=0;
14118
14119
14120 //function which gives binary length of n ,eg n=8->1000 length is 4
14121 int countBits(int n)
14122 {
14123     int count = 0;
14124     while (n)
14125     {
14126         count++;
14127         n >>= 1;
14128     }
14129     return count;
14130 }
14131
14132
14133 //binary representation of n
14134 string convertTobinary(int n)
14135 {
14136

```

```

14137     string b;
14138
14139     while(n)
14140     {
14141         if(n%2) b.pb('1');
14142         else b.pb('0');
14143         n=n/2;
14144     }
14145
14146     reverse(all(b));
14147     return b;
14148
14149 }
14150
14151
14152 void solve(){
14153
14154     int l,r;
14155     cin>>l>>r;
14156
14157     int length_l,length_r;
14158
14159     length_l=countBits(l);
14160     length_r=countBits(r);
14161
14162     if(length_l<length_r) // eg : l=1,r=3->11 we can make all zeroes except first digit we
get 10 so ans=1
14163     {
14164
14165
14166         cout<<length_r-1+(111<<(length_r))-1<<'\n';
14167         return;
14168
14169     }
14170
14171
14172
14173     string sl,sr; //binary representation of l and r
14174     int length=length_r;
14175
14176     sl=convertTobinary(l);
14177     sr=convertTobinary(r);
14178     int ans=0;
14179     for(int i=1;i<=min(40+l,r);i++)
14180     {
14181
14182         int j=i;
14183         int curr=0;
14184         int k=0;
14185         while(j)
14186         {
14187
14188             if(j==1)
14189             {
14190                 curr+=111<<k;
14191             }
14192             else if(j%2==0)
14193             {
14194                 curr+=111<<k;
14195             }
14196             j=j/2;
14197             k++;
14198
14199         }
14200         j=i;
14201         while(j)
14202         {
14203
14204             if(j%2==0) curr++;

```

```

14205     j=j/2;
14206 }
14207
14208 ans=max(ans,curr);
14209
14210 }
14211 cout<<ans<<'\n';
14212
14213 }
14214
14215 signed main(){
14216     ios_base::sync_with_stdio(false);
14217     cin.tie(0);cout.tie(0);
14218     //freopen("input.txt", "r", stdin);
14219     //freopen("output.txt", "w", stdout);
14220     #ifdef SIEVE
14221         sieve();
14222     #endif
14223     #ifdef NCR
14224         init();
14225     #endif
14226     int t=1;
14227     cin>>t;
14228     while(t--) solve();
14229     return 0;
14230 }
14231
14232 //PREFONES
14233 //Utkarsh.25dec
14234 #include <iostream>
14235 #include <cstdio>
14236 #include <cstdlib>
14237 #include <algorithm>
14238 #include <cmath>
14239 #include <vector>
14240 #include <set>
14241 #include <map>
14242 #include <unordered_set>
14243 #include <unordered_map>
14244 #include <queue>
14245 #include <ctime>
14246 #include <cassert>
14247 #include <complex>
14248 #include <string>
14249 #include <cstring>
14250 #include <chrono>
14251 #include <random>
14252 #include <bitset>
14253 #include <array>
14254 #define ll long long int
14255 #define pb push_back
14256 #define mp make_pair
14257 #define mod 1000000007
14258 #define vl vector<ll>
14259 #define all(c) (c).begin(),(c).end()
14260 using namespace std;
14261 ll power(ll a,ll b) {ll res=1;a%=mod; assert(b>=0); for(;b;b>>=1){if(b&1)res=res*a%mod;a=a*a%mod;}return res;}
14262 ll modInverse(ll a){return power(a,mod-2);}
14263 const int N=500023;
14264 bool vis[N];
14265 vector<int> adj[N];
14266 long long readInt(long long l,long long r,char endd){
14267     long long x=0;
14268     int cnt=0;
14269     int fi=-1;
14270     bool is_neg=false;
14271     while(true){
14272         char g=getchar();

```

```

14273         if(g=='-'){
14274             assert(fi==-1);
14275             is_neg=true;
14276             continue;
14277         }
14278         if('0'<=g && g<='9'){
14279             x*=10;
14280             x+=g-'0';
14281             if(cnt==0){
14282                 fi=g-'0';
14283             }
14284             cnt++;
14285             assert(fi!=0 || cnt==1);
14286             assert(fi!=0 || is_neg==false);
14287
14288             assert(!(cnt>19 || ( cnt==19 && fi>1) ));
14289         } else if(g==endd){
14290             if(is_neg){
14291                 x= -x;
14292             }
14293
14294             if(!(l <= x && x <= r))
14295             {
14296                 cerr << l << ' ' << r << ' ' << x << '\n';
14297                 assert(1 == 0);
14298             }
14299
14300             return x;
14301         } else {
14302             assert(false);
14303         }
14304     }
14305 }
14306 string readString(int l,int r,char endd){
14307     string ret="";
14308     int cnt=0;
14309     while(true){
14310         char g=getchar();
14311         assert(g!=-1);
14312         if(g==endd){
14313             break;
14314         }
14315         cnt++;
14316         ret+=g;
14317     }
14318     assert(l<=cnt && cnt<=r);
14319     return ret;
14320 }
14321 long long readIntSp(long long l,long long r){
14322     return readInt(l,r,' ');
14323 }
14324 long long readIntLn(long long l,long long r){
14325     return readInt(l,r,'\n');
14326 }
14327 string readStringLn(int l,int r){
14328     return readString(l,r,'\n');
14329 }
14330 string readStringSp(int l,int r){
14331     return readString(l,r,' ');
14332 }
14333 void solve()
14334 {
14335     int n=readInt(2,200000,'\n');
14336     string S=readString(n,n,'\n');
14337     for(auto ch:S)
14338         assert(ch=='0' || ch=='1');
14339     vector<int> cont;
14340     for(int i=0;i<n;i++)
14341     {

```

```

14342         if(S[i]=='1')
14343         {
14344             int j;
14345             int now=0;
14346             for(j=i;j<n;j++)
14347             {
14348                 if(S[j]=='1')
14349                     now++;
14350                 else
14351                     break;
14352             }
14353             cont.pb(now);
14354             i=j-1;
14355         }
14356     }
14357     if(S[0]=='0')
14358     {
14359         int ans=0;
14360         for(auto it:cont)
14361             ans=max(ans,it);
14362         cout<<ans<<'\n';
14363         return;
14364     }
14365     int ans=cont[0];
14366     int add=0;
14367     for(int i=1;i<cont.size();i++)
14368         add=max(add,cont[i]);
14369     ans+=add;
14370     cout<<ans<<'\n';
14371 }
14372 int main()
14373 {
14374     #ifndef ONLINE_JUDGE
14375     freopen("input.txt", "r", stdin);
14376     freopen("output.txt", "w", stdout);
14377     #endif
14378     ios_base::sync_with_stdio(false);
14379     cin.tie(NULL),cout.tie(NULL);
14380     int T=readInt(1,1000,'\n');
14381     while(T--)
14382         solve();
14383     assert(getchar()==-1);
14384     cerr << "Time : " << 1000 * ((double)clock()) / (double)CLOCKS_PER_SEC << "ms\n";
14385 }
14386
14387 //SUMRANGEPOW
14388 #include "bits/stdc++.h"
14389 // #pragma GCC optimize("O3,unroll-loops")
14390 // #pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
14391 using namespace std;
14392 using ll = long long int;
14393 mt19937_64 rng(chrono::high_resolution_clock::now().time_since_epoch().count());
14394
14395 const int mod = 998244353;
14396 int add(int a, int b) {
14397     return (a + b) % mod;
14398 }
14399 int mul(int a, int b) {
14400     return (1LL * a * b) % mod;
14401 }
14402
14403 vector C(100, vector(100, 0));
14404 vector pows(15, vector(100005, 0));
14405
14406 struct Node {
14407     static const int kmax = 6;
14408     using T = array<int, kmax>;
14409     T unit {};
14410     T f(T a, T b) {

```

```

14411         for (int i = 0; i < kmax; ++i) a[i] = ::add(a[i], b[i]);
14412         return a;
14413     }
14414
14415     Node *l = 0, *r = 0;
14416     int lo, hi;
14417     int madd = 0;
14418     T val = unit;
14419     Node(int _lo, int _hi):lo(_lo),hi(_hi){ val[0] = hi - lo; }
14420     T query(int L, int R) {
14421         if (R <= lo || hi <= L) return unit;
14422         if (L <= lo && hi <= R) return val;
14423         push();
14424         return f(l->query(L, R), r->query(L, R));
14425     }
14426     void add(int L, int R, int x) {
14427         if (R <= lo || hi <= L) return;
14428         if (L <= lo && hi <= R) {
14429             madd += x;
14430             for (int k = kmax-1; k > 0; --k) {
14431                 int cur = 0;
14432                 for (int j = k; j >= 0; --j) {
14433                     cur = ::add(cur, mul(val[j], mul(C[k][j], pows[k-j][x])));
14434                 }
14435                 val[k] = cur;
14436             }
14437         }
14438         else {
14439             push(), l->add(L, R, x), r->add(L, R, x);
14440             val = f(l->val, r->val);
14441         }
14442     }
14443     void push() {
14444         if (!l) {
14445             int mid = lo + (hi - lo)/2;
14446             l = new Node(lo, mid); r = new Node(mid, hi);
14447         }
14448         if (madd)
14449             l->add(lo,hi,madd), r->add(lo,hi,madd), madd = 0;
14450     }
14451 };
14452
14453 int main()
14454 {
14455     ios::sync_with_stdio(false); cin.tie(0);
14456
14457     for (int i = 0; i < 100; ++i) C[i][0] = 1;
14458     for (int i = 1; i < 100; ++i) for (int j = 1; j <= i; ++j)
14459         C[i][j] = add(C[i-1][j], C[i-1][j-1]);
14460
14461     for (int i = 1; i < 100005; ++i) for (int j = 0; j < 12; ++j)
14462         if (j == 0) pows[j][i] = 1;
14463         else pows[j][i] = mul(i, pows[j-1][i]);
14464
14465     int t; cin >> t;
14466     while (t--) {
14467         int n, k; cin >> n >> k;
14468         vector<int> a(n);
14469         for (int &x : a) cin >> x;
14470         vector<int> prev(n+1, -1);
14471         Node *seg = new Node(0, n);
14472
14473         int ans = 0;
14474         for (int R = 0; R < n; ++R) {
14475             seg -> add(prev[a[R]]+1, R+1, 1);
14476             ans = add(ans, (seg -> query(0, n))[k]);
14477             prev[a[R]] = R;
14478         }
14479         cout << ans << '\n';

```

```

14480     }
14481 }
14482
14483 //MISREP
14484 #include<bits/stdc++.h>
14485 #include<string>
14486
14487 using namespace std;
14488
14489 #define ll long long int
14490 #define ld long double
14491 #define pb push_back
14492 #define all(v) v.begin(),v.end()
14493 #define sz(x) ((int)(x).size())
14494 #define deb(x) cout<< #x << '=' << x <<endl
14495 #define MOD 1000000007
14496 const int N = 501;
14497 ll n;
14498 ll a[N] , cache[N][N * N];
14499
14500 ll dp(ll i , ll s){
14501     if(s < 0){
14502         return 0;
14503     }
14504     if(i >= n){
14505         if(s == 0){
14506             cache[i][s] = 1;
14507             return 1;
14508         }
14509         cache[i][s] = 0;
14510         return 0;
14511     }
14512
14513     ll &ans = cache[i][s];
14514     if(ans != -1){
14515         return ans;
14516     }
14517
14518     ll res = 0;
14519     res |= dp(i + 1 , s - a[i]);
14520     res |= dp(i + 1 , s);
14521     return ans = res;
14522 }
14523
14524 int main() {
14525
14526     ll t;
14527     cin>>t;
14528
14529     assert(t <= 50 && t >= 1);
14530
14531     ll sumN = 0;
14532     for(int tc = 0; tc < t; tc++){
14533         cin>>n;
14534
14535         sumN += n;
14536
14537         assert(n >= 2 && n <= 300);
14538
14539         ll sum = 0;
14540         for(int i = 0; i < n; i++){
14541             cin>>a[i];
14542             assert(a[i] >= 1 && a[i] <= 300);
14543             sum += a[i];
14544         }
14545
14546         if(sum % 2){
14547             cout<<-1<<"\n";
14548             continue;

```

```

14549     }
14550
14551     for(int i = 0; i <= n; i++){
14552         for(int j = 0; j <= sum; j++){
14553             cache[i][j] = -1;
14554         }
14555     }
14556
14557     ll flag = dp(0 , sum/2);
14558
14559     if(flag){
14560         set<ll> s1 , s2;
14561         for(int i = 0; i < n; i++){
14562             s2.insert(i);
14563         }
14564
14565         ll i = 0 , s = sum/2;
14566         while(i < n){
14567             if(cache[i + 1][s - a[i]]){
14568                 s -= a[i];
14569                 s1.insert(i);
14570                 s2.erase(i);
14571             }
14572             i++;
14573         }
14574
14575         vector<pair<ll , ll>> ans;
14576         while(sz(s1)){
14577             ll idx1 = *s1.begin() , idx2 = *s2.begin();
14578             ll mn = min(a[idx1] , a[idx2]);
14579             a[idx1] -= mn;
14580             a[idx2] -= mn;
14581
14582             ans.pb({idx1 + 1 , idx2 + 1});
14583             if(a[idx1] == 0){
14584                 s1.erase(idx1);
14585             }
14586
14587             if(a[idx2] == 0){
14588                 s2.erase(idx2);
14589             }
14590         }
14591
14592         // assert(sz(ans) >= 0 && sz(ans) <= n);
14593         cout<<sz(ans)<<"\n";
14594         for(auto it : ans){
14595             cout<<it.first<<" "<<it.second<<"\n";
14596         }
14597     }else{
14598         cout<<-1<<"\n";
14599     }
14600 }
14601
14602     assert(sumN <= 600);
14603     return 0;
14604 }
14605
14606 //XOR_X
14607 #include <map>
14608 #include <set>
14609 #include <cmath>
14610 #include <ctime>
14611 #include <queue>
14612 #include <stack>
14613 #include <cstdio>
14614 #include <cstdlib>
14615 #include <vector>
14616 #include <cstring>
14617 #include <algorithm>

```



```

14618 #include <iostream>
14619 using namespace std;
14620 typedef double db;
14621 typedef long long ll;
14622 typedef unsigned long long ull;
14623 const int N=1000010;
14624 const int LOGN=29;
14625 const ll TMD=0;
14626 const ll INF=2147483647;
14627 int n,last_max,cur_max;
14628
14629 void work(int bt)
14630 {
14631     for(int i=1;i<=n;i++)
14632     {
14633         printf("%d %d\n",i,(1<<bt));
14634         fflush(stdout);
14635         scanf("%d",&cur_max);
14636         if(cur_max==0)
14637         {
14638             printf("0\n");
14639             fflush(stdout);
14640             scanf("%d",&cur_max);
14641             return ;
14642         }
14643         if(cur_max!=last_max)
14644         {
14645             printf("%d %d\n",i,cur_max);
14646             fflush(stdout);
14647             scanf("%d",&cur_max);
14648             if(cur_max==0)
14649             {
14650                 printf("0\n");
14651                 fflush(stdout);
14652                 scanf("%d",&cur_max);
14653                 return ;
14654             }
14655             last_max=cur_max;
14656         }
14657     }
14658 }
14659
14660 int main()
14661 {
14662     scanf("%d",&n);
14663     printf("1 0\n");
14664     fflush(stdout);
14665     scanf("%d",&cur_max);
14666     last_max=cur_max;
14667     for(int i=0;i<=LOGN;i++)
14668     {
14669         work(i);
14670         if(!i) work(i);
14671     }
14672
14673     return 0;
14674 }
14675
14676 //CANDIES3
14677 #include <bits/stdc++.h>
14678 #include "stdio.h"
14679
14680 using namespace std;
14681
14682 #define SZ(s) ((int)s.size())
14683 #define all(x) (x).begin(), (x).end()
14684 #define lla(x) (x).rbegin(), (x).rend()
14685 #define bpc(x) __builtin_popcount(x)
14686 #define bpc11(x) __builtin_popcountll(x)

```

```

14687 #define MP make_pair
14688 #define endl '\n'
14689
14690 mt19937 rng(chrono::high_resolution_clock::now().time_since_epoch().count());
14691
14692 typedef long long ll;
14693 const int MOD = 1e9 + 7;
14694 const int N = 1e6 + 3e2;
14695
14696 int sumn = 0, summ = 0;
14697 void solve(){
14698     int n, m;
14699     cin >> n >> m;
14700
14701     sumn += n;
14702     summ += m;
14703
14704     vector<int> c(m + 1), cnt(m + 1, 0);
14705     while (n--){
14706         int x;
14707         cin >> x;
14708         assert(1 <= x && x <= m);
14709         cnt[x]++;
14710     }
14711
14712     for (int i = 1; i <= m; i++){
14713         cin >> c[i];
14714         assert(1 <= c[i] && c[i] <= 1000000);
14715     }
14716
14717     for (int i = 2; i <= m; i++) cnt[i] += cnt[i - 1];
14718
14719     long long ans = 0;
14720     for (int p = 1; p <= m; p++){
14721         long long candies = 0;
14722         for (int x = 1; x <= m / p; x++){
14723             int l = x * p, r = min(m, (x + 1) * p - 1);
14724             candies += (ll)(cnt[r] - cnt[l - 1]) * x;
14725         }
14726         ans = max(ans, candies * c[p]);
14727     }
14728
14729     cout << ans << endl;
14730 }
14731
14732 int main(){
14733     clock_t startTime = clock();
14734     ios_base::sync_with_stdio(false);
14735
14736 #ifdef LOCAL
14737     freopen("input.txt", "r", stdin);
14738     freopen("output.txt", "w", stdout);
14739     freopen("error.txt", "w", stderr);
14740 #endif
14741
14742     int test_cases = 1;
14743     cin >> test_cases;
14744
14745     assert(1 <= test_cases && test_cases <= 10000);
14746
14747     for (int test = 1; test <= test_cases; test++){
14748         // cout << (solve() ? "YES" : "NO") << endl;
14749         solve();
14750     }
14751
14752     assert(sumn <= 100000);
14753     assert(summ <= 100000);
14754
14755     cerr << "Time: " << int((double) (clock() - startTime) / CLOCKS_PER_SEC * 1000) <<

```

```
" ms" << endl;
```

```
return 0;
```

```
}
```

```
//DIST_VALS
```

```
#include "bits/stdc++.h"
```

```
// #include "testlib.h"
```

```
using namespace std;
```

```
// #include <ext/pb_ds/assoc_container.hpp>
```

```
// using namespace __gnu_pbds;
```

```
// template<class T> using oset = tree<T,null_type,less_equal// for indexed_multiset */  
// <T> ,rb_tree_tag,tree_order_statistics_node_update> ; // order_of_key(k) -> # of  
elem strictly < k .
```

```
// // *(s.find_by_order(k)) ->  
element at index K .
```

```
#define int long long int
```

```
using ll= long long;
```

```
#define ld long double
```

```
#define endl '\n'
```

```
#define dbg(x) cout<<#x<<" is -> "<<x<<endl
```

```
#define speed_ ios_base::sync_with_stdio(false),cin.tie(0), cout.tie(0)
```

```
#define pb push_back
```

```
#define po pop_back
```

```
#define mp make_pair
```

```
#define sab(x) x.begin(),x.end()
```

```
#define rsab(x) x.rbegin(),x.rend()
```

```
#define ff first
```

```
#define ss second
```

```
#define sz(x) (int)x.size()
```

```
#define sp(x) fixed<<setprecision(x)
```

```
#define uni(edge) edge.erase(unique(edge.begin(),edge.end()),edge.end());
```

```
#define to_up(x) transform(sab(x),x.begin(),::toupper)
```

```
#define to_low(x) transform(x.begin(),x.end(),x.begin(),::tolower)
```

```
#define ONLINE_JUDGE
```

```
// const int M = 1000000007;
```

```
// const int MM = 998244353;
```

```
// const ld Pi= acos(-1);
```

```
// const int N=1e5+10;
```

```
// const int inf=1e18;
```

```
// const int MAXX=1e9;
```

```
vector<int>v;
```

```
int t;
```

```
int test_count=0;
```

```
void simp(){
```

```
// dp?, graph?, bs on answer?, compress/sort queries/array?, stupid observation?
```

```
test_count++;
```

```
int n;
```

```
cin>>n;
```

```
v.resize(n);
```

```
set<int>s;
```

```
stack<int>st1;
```

```
for(int i=0;i<n;i++){
```

```
cin>>v[i];
```

```
}
```

```
for(int i=0;i<n;i++){
```

```
while(st1.size() && st1.top()<=v[i]){
```

```
int curr=st1.top();
```

```
s.insert(v[i]-curr);
```

```
st1.pop();
```

```
}
```

```
st1.push(v[i]);
```

```
}
```

```

14822     while(sz(stl)){
14823         stl.pop();
14824     }
14825     reverse(sab(v));
14826     for(int i=0;i<n;i++){
14827         while(stl.size() && stl.top()<=v[i]){
14828             int curr=stl.top();
14829             s.insert(v[i]-curr);
14830             stl.pop();
14831         }
14832         stl.push(v[i]);
14833     }
14834
14835     int ans=sz(s);
14836     cout<<ans;
14837     if(test_count!=t){
14838         cout<<endl;
14839     }
14840
14841 }
14842
14843 signed main(){
14844
14845     speed_;// remove this in interactive problems
14846
14847     // freopen("ouput05.txt", "r", stdin);
14848     // freopen("input05.txt", "w", stdout);
14849
14850     // int t;
14851     t=1;
14852     cin>>t;
14853
14854     // initialize();
14855     // solve();
14856
14857     //gen_factorial(N+10);
14858
14859     int curr=1;
14860     for(int i=0;i<t;i++){
14861
14862         #ifndef ONLINE_JUDGE
14863
14864         #endif
14865         // cout<<"Case #"<<curr++<<": ";
14866         simp();
14867
14868     }
14869     return 0;
14870 }
14871
14872
14873 //MEDSTRONG
14874 #pragma GCC optimize("O3,unroll-loops")
14875 #pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
14876
14877 #include <bits/stdc++.h>
14878 using namespace std;
14879
14880 // #include <ext/pb_ds/assoc_container.hpp> //required
14881 // #include <ext/pb_ds/tree_policy.hpp> //required
14882 // using namespace __gnu_pbds; //required
14883 // template <typename T> using ordered_set = tree<T, null_type, less<T>,
14884 // rb_tree_tag, tree_order_statistics_node_update>;
14885
14886 // ordered_set <int> s;
14887 // s.find_by_order(k); returns the (k+1)th smallest element
14888 // s.order_of_key(k); returns the number of elements in s strictly less than k
14889
14890 #define pb push_back

```

```

14891 #define mp(x, y) make_pair(x, y)
14892 #define all(x) x.begin(), x.end()
14893 #define allr(x) x.rbegin(), x.rend()
14894 #define leftmost_bit(x) (63 - __builtin_clzll(x))
14895 #define rightmost_bit(x) __builtin_ctzll(x) // count trailing zeros
14896 #define set_bits(x) __builtin_popcountll(x)
14897 #define pow2(i) (1LL << (i))
14898 #define is_on(x, i) ((x)&pow2(i)) // state of the ith bit in x
14899 #define set_on(x, i) ((x) | pow2(i)) // returns integer x with ith bit on
14900 #define set_off(x, i) ((x) & ~pow2(i)) // returns integer x with ith bit off
14901 #define fi first
14902 #define se second
14903
14904 typedef long long int ll;
14905 typedef long double ld;
14906
14907 const int MOD = 1e9 + 7; // 998244353;
14908 const int MX = 2e5 + 5;
14909 const ll INF = 1e18; // not too close to LLONG_MAX
14910 const ld PI = acos((ld)-1);
14911 const ld EPS = 1e-8;
14912 const int dx[4] = {1, 0, -1, 0},
14913             dy[4] = {0, 1, 0, -1}; // for every grid problem!!
14914
14915 // hash map and operator overload from
14916 // https://www.youtube.com/watch?v=jkfA0Ts6YBA Custom hash map
14917 struct custom_hash {
14918     static uint64_t splitmix64(uint64_t x) {
14919         x += 0x9e3779b97f4a7c15;
14920         x = (x ^ (x >> 30)) * 0xbf58476d1ce4e5b9;
14921         x = (x ^ (x >> 27)) * 0x94d049bb133111eb;
14922         return x ^ (x >> 31);
14923     }
14924
14925     size_t operator()(uint64_t x) const {
14926         static const uint64_t FIXED_RANDOM =
14927             chrono::steady_clock::now().time_since_epoch().count();
14928         return splitmix64(x + FIXED_RANDOM);
14929     }
14930 };
14931 template <typename T1, typename T2> // Key should be integer type
14932 using safe_map = unordered_map<T1, T2, custom_hash>;
14933
14934 // Operator overloads
14935 template <typename T1, typename T2> // cin >> pair<T1, T2>
14936 istream &operator>>(istream &istream, pair<T1, T2> &p) {
14937     return (istream >> p.first >> p.second);
14938 }
14939 template <typename T1, typename T2> // cout << pair<T1, T2>
14940 ostream &operator<<(ostream &ostream, const pair<T1, T2> &p) {
14941     return (ostream << p.first << " " << p.second);
14942 }
14943
14944 template <typename T> // cin >> array<T, 2>
14945 istream &operator>>(istream &istream, array<T, 2> &p) {
14946     return (istream >> p[0] >> p[1]);
14947 }
14948 template <typename T> // cout << array<T, 2>
14949 ostream &operator<<(ostream &ostream, const array<T, 2> &p) {
14950     return (ostream << p[0] << " " << p[1]);
14951 }
14952
14953 template <typename T> // cin >> vector<T>
14954 istream &operator>>(istream &istream, vector<T> &v) {
14955     for (auto &it : v)
14956         cin >> it;
14957     return istream;
14958 }
14959 template <typename T> // cout << vector<T>

```

```

14960 ostream &operator<<(ostream &ostream, const vector<T> &c) {
14961     for (auto &it : c)
14962         cout << it << " ";
14963     return ostream;
14964 }
14965 clock_t startTime;
14966 mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());
14967 double getCurrentTime() {
14968     return (double)(clock() - startTime) / CLOCKS_PER_SEC;
14969 }
14970 string to_string(string s) { return '"' + s + '"'; }
14971 string to_string(const char *s) { return to_string((string)s); }
14972 string to_string(bool b) { return (b ? "true" : "false"); }
14973 int getRandomNumber(int l, int r) {
14974     uniform_int_distribution<int> dist(l, r);
14975     return dist(rng);
14976 }
14977
14978 // https://github.com/the-tourist/algo/blob/master/misc/debug.cpp
14979 template <typename A, typename B> string to_string(pair<A, B> p) {
14980     return "(" + to_string(p.first) + ", " + to_string(p.second) + ")";
14981 }
14982 template <typename A> string to_string(A v) {
14983     bool first = true;
14984     string res = "{";
14985     for (const auto &x : v) {
14986         if (!first) {
14987             res += ", ";
14988         }
14989         first = false;
14990         res += to_string(x);
14991     }
14992     res += "}";
14993     return res;
14994 }
14995 void debug_out() { cerr << endl; }
14996 template <typename Head, typename... Tail> void debug_out(Head H, Tail... T) {
14997     cerr << " " << to_string(H);
14998     debug_out(T...);
14999 }
15000
15001 #ifdef LOCAL_DEBUG
15002 #define debug(...) cerr << "[" << #__VA_ARGS__ << "]:", debug_out(__VA_ARGS__)
15003 #else
15004 #define debug(...) ;
15005 #endif
15006
15007 // #define int ll // disable when you want to use atcoder library
15008 #define endl '\n' // disable when dealing with interactive problems
15009
15010 typedef vector<int> vi;
15011 typedef pair<int, int> pii;
15012 typedef array<int, 2>
15013     edge; // for graphs, make it array<int,3> for weighted edges
15014
15015
15016 #include <cassert>
15017 #include <numeric>
15018 #include <type_traits>
15019
15020 #ifdef _MSC_VER
15021 #include <intrin.h>
15022 #endif
15023
15024
15025 #include <utility>
15026
15027 #ifdef _MSC_VER
15028 #include <intrin.h>

```

```

15029 #endif
15030
15031 namespace atcoder {
15032
15033 namespace internal {
15034
15035 constexpr long long safe_mod(long long x, long long m) {
15036     x %= m;
15037     if (x < 0) x += m;
15038     return x;
15039 }
15040
15041 struct barrett {
15042     unsigned int _m;
15043     unsigned long long im;
15044
15045     explicit barrett(unsigned int m) : _m(m), im((unsigned long long)(-1) / m + 1) {}
15046
15047     unsigned int umod() const { return _m; }
15048
15049     unsigned int mul(unsigned int a, unsigned int b) const {
15050
15051         unsigned long long z = a;
15052         z *= b;
15053 #ifdef _MSC_VER
15054         unsigned long long x;
15055         _umul128(z, im, &x);
15056 #else
15057         unsigned long long x =
15058             (unsigned long long)(((unsigned __int128)(z)*im) >> 64);
15059 #endif
15060         unsigned int v = (unsigned int)(z - x * _m);
15061         if (_m <= v) v += _m;
15062         return v;
15063     }
15064 };
15065
15066 constexpr long long pow_mod_constexpr(long long x, long long n, int m) {
15067     if (m == 1) return 0;
15068     unsigned int _m = (unsigned int)(m);
15069     unsigned long long r = 1;
15070     unsigned long long y = safe_mod(x, m);
15071     while (n) {
15072         if (n & 1) r = (r * y) % _m;
15073         y = (y * y) % _m;
15074         n >>= 1;
15075     }
15076     return r;
15077 }
15078
15079 constexpr bool is_prime_constexpr(int n) {
15080     if (n <= 1) return false;
15081     if (n == 2 || n == 7 || n == 61) return true;
15082     if (n % 2 == 0) return false;
15083     long long d = n - 1;
15084     while (d % 2 == 0) d /= 2;
15085     constexpr long long bases[3] = {2, 7, 61};
15086     for (long long a : bases) {
15087         long long t = d;
15088         long long y = pow_mod_constexpr(a, t, n);
15089         while (t != n - 1 && y != 1 && y != n - 1) {
15090             y = y * y % n;
15091             t <<= 1;
15092         }
15093         if (y != n - 1 && t % 2 == 0) {
15094             return false;
15095         }
15096     }
15097     return true;

```

```

15098 }
15099 template <int n> constexpr bool is_prime = is_prime_constexpr(n);
15100
15101 constexpr std::pair<long long, long long> inv_gcd(long long a, long long b) {
15102     a = safe_mod(a, b);
15103     if (a == 0) return {b, 0};
15104
15105     long long s = b, t = a;
15106     long long m0 = 0, m1 = 1;
15107
15108     while (t) {
15109         long long u = s / t;
15110         s -= t * u;
15111         m0 -= m1 * u; // |m1 * u| <= |m1| * s <= b
15112
15113         auto tmp = s;
15114         s = t;
15115         t = tmp;
15116         tmp = m0;
15117         m0 = m1;
15118         m1 = tmp;
15119     }
15120     if (m0 < 0) m0 += b / s;
15121     return {s, m0};
15122 }
15123
15124
15125 constexpr int primitive_root_constexpr(int m) {
15126     if (m == 2) return 1;
15127     if (m == 167772161) return 3;
15128     if (m == 469762049) return 3;
15129     if (m == 754974721) return 11;
15130     if (m == 998244353) return 3;
15131     int divs[20] = {};
15132     divs[0] = 2;
15133     int cnt = 1;
15134     int x = (m - 1) / 2;
15135     while (x % 2 == 0) x /= 2;
15136     for (int i = 3; (long long)(i)*i <= x; i += 2) {
15137         if (x % i == 0) {
15138             divs[cnt++] = i;
15139             while (x % i == 0) {
15140                 x /= i;
15141             }
15142         }
15143     }
15144     if (x > 1) {
15145         divs[cnt++] = x;
15146     }
15147     for (int g = 2;; g++) {
15148         bool ok = true;
15149         for (int i = 0; i < cnt; i++) {
15150             if (pow_mod_constexpr(g, (m - 1) / divs[i], m) == 1) {
15151                 ok = false;
15152                 break;
15153             }
15154         }
15155         if (ok) return g;
15156     }
15157 }
15158 template <int m> constexpr int primitive_root = primitive_root_constexpr(m);
15159
15160 unsigned long long floor_sum_unsigned(unsigned long long n,
15161                                     unsigned long long m,
15162                                     unsigned long long a,
15163                                     unsigned long long b) {
15164     unsigned long long ans = 0;
15165     while (true) {
15166         if (a >= m) {

```



[illegible]

```

15236         is_unsigned_int128<T>::value,
15237         std::true_type,
15238         std::false_type>::type;
15239
15240 template <class T>
15241 using to_unsigned = typename std::conditional<
15242     is_signed_int128<T>::value,
15243     make_unsigned_int128<T>,
15244     typename std::conditional<std::is_signed<T>::value,
15245         std::make_unsigned<T>,
15246         std::common_type<T>>::type>::type;
15247
15248 #else
15249
15250 template <class T> using is_integral = typename std::is_integral<T>;
15251
15252 template <class T>
15253 using is_signed_int =
15254     typename std::conditional<is_integral<T>::value && std::is_signed<T>::value,
15255         std::true_type,
15256         std::false_type>::type;
15257
15258 template <class T>
15259 using is_unsigned_int =
15260     typename std::conditional<is_integral<T>::value &&
15261         std::is_unsigned<T>::value,
15262         std::true_type,
15263         std::false_type>::type;
15264
15265 template <class T>
15266 using to_unsigned = typename std::conditional<is_signed_int<T>::value,
15267     std::make_unsigned<T>,
15268     std::common_type<T>>::type;
15269
15270 #endif
15271
15272 template <class T>
15273 using is_signed_int_t = std::enable_if_t<is_signed_int<T>::value>;
15274
15275 template <class T>
15276 using is_unsigned_int_t = std::enable_if_t<is_unsigned_int<T>::value>;
15277
15278 template <class T> using to_unsigned_t = typename to_unsigned<T>::type;
15279
15280 } // namespace internal
15281
15282 } // namespace atcoder
15283
15284
15285 namespace atcoder {
15286
15287     namespace internal {
15288
15289         struct modint_base {};
15290         struct static_modint_base : modint_base {};
15291
15292         template <class T> using is_modint = std::is_base_of<modint_base, T>;
15293         template <class T> using is_modint_t = std::enable_if_t<is_modint<T>::value>;
15294
15295     } // namespace internal
15296
15297     template <int m, std::enable_if_t<(1 <= m)>* = nullptr>
15298     struct static_modint : internal::static_modint_base {
15299         using mint = static_modint;
15300
15301     public:
15302         static constexpr int mod() { return m; }
15303         static mint raw(int v) {
15304             mint x;

```

```

15305         x._v = v;
15306         return x;
15307     }
15308
15309     static_modint() : _v(0) {}
15310     template <class T, internal::is_signed_int_t<T>* = nullptr>
15311     static_modint(T v) {
15312         long long x = (long long)(v % (long long)(umod()));
15313         if (x < 0) x += umod();
15314         _v = (unsigned int)(x);
15315     }
15316     template <class T, internal::is_unsigned_int_t<T>* = nullptr>
15317     static_modint(T v) {
15318         _v = (unsigned int)(v % umod());
15319     }
15320
15321     unsigned int val() const { return _v; }
15322
15323     mint& operator++() {
15324         _v++;
15325         if (_v == umod()) _v = 0;
15326         return *this;
15327     }
15328     mint& operator--() {
15329         if (_v == 0) _v = umod();
15330         _v--;
15331         return *this;
15332     }
15333     mint operator++(int) {
15334         mint result = *this;
15335         ++*this;
15336         return result;
15337     }
15338     mint operator--(int) {
15339         mint result = *this;
15340         --*this;
15341         return result;
15342     }
15343
15344     mint& operator+=(const mint& rhs) {
15345         _v += rhs._v;
15346         if (_v >= umod()) _v -= umod();
15347         return *this;
15348     }
15349     mint& operator-=(const mint& rhs) {
15350         _v -= rhs._v;
15351         if (_v >= umod()) _v += umod();
15352         return *this;
15353     }
15354     mint& operator*=(const mint& rhs) {
15355         unsigned long long z = _v;
15356         z *= rhs._v;
15357         _v = (unsigned int)(z % umod());
15358         return *this;
15359     }
15360     mint& operator/=(const mint& rhs) { return *this = *this * rhs.inv(); }
15361
15362     mint operator+() const { return *this; }
15363     mint operator-() const { return mint() - *this; }
15364
15365     mint pow(long long n) const {
15366         assert(0 <= n);
15367         mint x = *this, r = 1;
15368         while (n) {
15369             if (n & 1) r *= x;
15370             x *= x;
15371             n >>= 1;
15372         }
15373         return r;

```

```

15374     }
15375     mint inv() const {
15376         if (prime) {
15377             assert(_v);
15378             return pow(umod() - 2);
15379         } else {
15380             auto eg = internal::inv_gcd(_v, m);
15381             assert(eg.first == 1);
15382             return eg.second;
15383         }
15384     }
15385
15386     friend mint operator+(const mint& lhs, const mint& rhs) {
15387         return mint(lhs) += rhs;
15388     }
15389     friend mint operator-(const mint& lhs, const mint& rhs) {
15390         return mint(lhs) -= rhs;
15391     }
15392     friend mint operator*(const mint& lhs, const mint& rhs) {
15393         return mint(lhs) *= rhs;
15394     }
15395     friend mint operator/(const mint& lhs, const mint& rhs) {
15396         return mint(lhs) /= rhs;
15397     }
15398     friend bool operator==(const mint& lhs, const mint& rhs) {
15399         return lhs._v == rhs._v;
15400     }
15401     friend bool operator!=(const mint& lhs, const mint& rhs) {
15402         return lhs._v != rhs._v;
15403     }
15404
15405 private:
15406     unsigned int _v;
15407     static constexpr unsigned int umod() { return m; }
15408     static constexpr bool prime = internal::is_prime<m>;
15409 };
15410
15411 template <int id> struct dynamic_modint : internal::modint_base {
15412     using mint = dynamic_modint;
15413
15414 public:
15415     static int mod() { return (int)(bt.umod()); }
15416     static void set_mod(int m) {
15417         assert(1 <= m);
15418         bt = internal::barrett(m);
15419     }
15420     static mint raw(int v) {
15421         mint x;
15422         x._v = v;
15423         return x;
15424     }
15425
15426     dynamic_modint() : _v(0) {}
15427     template <class T, internal::is_signed_int_t<T>* = nullptr>
15428     dynamic_modint(T v) {
15429         long long x = (long long)(v % (long long)(mod()));
15430         if (x < 0) x += mod();
15431         _v = (unsigned int)(x);
15432     }
15433     template <class T, internal::is_unsigned_int_t<T>* = nullptr>
15434     dynamic_modint(T v) {
15435         _v = (unsigned int)(v % mod());
15436     }
15437
15438     unsigned int val() const { return _v; }
15439
15440     mint& operator++() {
15441         _v++;
15442         if (_v == umod()) _v = 0;

```

```

15443         return *this;
15444     }
15445     mint& operator--() {
15446         if (_v == 0) _v = umod();
15447         _v--;
15448         return *this;
15449     }
15450     mint operator++(int) {
15451         mint result = *this;
15452         ++*this;
15453         return result;
15454     }
15455     mint operator--(int) {
15456         mint result = *this;
15457         --*this;
15458         return result;
15459     }
15460
15461     mint& operator+=(const mint& rhs) {
15462         _v += rhs._v;
15463         if (_v >= umod()) _v -= umod();
15464         return *this;
15465     }
15466     mint& operator-=(const mint& rhs) {
15467         _v += mod() - rhs._v;
15468         if (_v >= umod()) _v -= umod();
15469         return *this;
15470     }
15471     mint& operator*=(const mint& rhs) {
15472         _v = bt.mul(_v, rhs._v);
15473         return *this;
15474     }
15475     mint& operator/=(const mint& rhs) { return *this = *this * rhs.inv(); }
15476
15477     mint operator+() const { return *this; }
15478     mint operator-() const { return mint() - *this; }
15479
15480     mint pow(long long n) const {
15481         assert(0 <= n);
15482         mint x = *this, r = 1;
15483         while (n) {
15484             if (n & 1) r *= x;
15485             x *= x;
15486             n >>= 1;
15487         }
15488         return r;
15489     }
15490     mint inv() const {
15491         auto eg = internal::inv_gcd(_v, mod());
15492         assert(eg.first == 1);
15493         return eg.second;
15494     }
15495
15496     friend mint operator+(const mint& lhs, const mint& rhs) {
15497         return mint(lhs) += rhs;
15498     }
15499     friend mint operator-(const mint& lhs, const mint& rhs) {
15500         return mint(lhs) -= rhs;
15501     }
15502     friend mint operator*(const mint& lhs, const mint& rhs) {
15503         return mint(lhs) *= rhs;
15504     }
15505     friend mint operator/(const mint& lhs, const mint& rhs) {
15506         return mint(lhs) /= rhs;
15507     }
15508     friend bool operator==(const mint& lhs, const mint& rhs) {
15509         return lhs._v == rhs._v;
15510     }
15511     friend bool operator!=(const mint& lhs, const mint& rhs) {

```

```

15512         return lhs._v != rhs._v;
15513     }
15514
15515     private:
15516         unsigned int _v;
15517         static internal::barrett bt;
15518         static unsigned int umod() { return bt.umod(); }
15519 };
15520 template <int id> internal::barrett dynamic_modint<id>::bt(998244353);
15521
15522 using modint998244353 = static_modint<998244353>;
15523 using modint1000000007 = static_modint<1000000007>;
15524 using modint = dynamic_modint<-1>;
15525
15526 namespace internal {
15527
15528     template <class T>
15529     using is_static_modint = std::is_base_of<internal::static_modint_base, T>;
15530
15531     template <class T>
15532     using is_static_modint_t = std::enable_if_t<is_static_modint<T>::value>;
15533
15534     template <class> struct is_dynamic_modint : public std::false_type {};
15535     template <int id>
15536     struct is_dynamic_modint<dynamic_modint<id>> : public std::true_type {};
15537
15538     template <class T>
15539     using is_dynamic_modint_t = std::enable_if_t<is_dynamic_modint<T>::value>;
15540
15541 } // namespace internal
15542
15543 } // namespace atcoder
15544
15545
15546 #include <algorithm>
15547 #include <cassert>
15548 #include <vector>
15549
15550
15551 #include <algorithm>
15552 #include <utility>
15553 #include <vector>
15554
15555
15556 #include <algorithm>
15557 #include <utility>
15558 #include <vector>
15559
15560 namespace atcoder {
15561 namespace internal {
15562
15563     template <class E> struct csr {
15564         std::vector<int> start;
15565         std::vector<E> elist;
15566         explicit csr(int n, const std::vector<std::pair<int, E>>& edges)
15567             : start(n + 1), elist(edges.size()) {
15568             for (auto e : edges) {
15569                 start[e.first + 1]++;
15570             }
15571             for (int i = 1; i <= n; i++) {
15572                 start[i] += start[i - 1];
15573             }
15574             auto counter = start;
15575             for (auto e : edges) {
15576                 elist[counter[e.first]++] = e.second;
15577             }
15578         }
15579 };
15580

```

```

15581 } // namespace internal
15582
15583 } // namespace atcoder
15584
15585
15586 namespace atcoder {
15587 namespace internal {
15588
15589 struct scc_graph {
15590 public:
15591     explicit scc_graph(int n) : _n(n) {}
15592
15593     int num_vertices() { return _n; }
15594
15595     void add_edge(int from, int to) { edges.push_back({from, to}); }
15596
15597     std::pair<int, std::vector<int>> scc_ids() {
15598         auto g = csr<edge>(_n, edges);
15599         int now_ord = 0, group_num = 0;
15600         std::vector<int> visited, low(_n), ord(_n, -1), ids(_n);
15601         visited.reserve(_n);
15602         auto dfs = [&](auto self, int v) -> void {
15603             low[v] = ord[v] = now_ord++;
15604             visited.push_back(v);
15605             for (int i = g.start[v]; i < g.start[v + 1]; i++) {
15606                 auto to = g.elist[i].to;
15607                 if (ord[to] == -1) {
15608                     self(self, to);
15609                     low[v] = std::min(low[v], low[to]);
15610                 } else {
15611                     low[v] = std::min(low[v], ord[to]);
15612                 }
15613             }
15614             if (low[v] == ord[v]) {
15615                 while (true) {
15616                     int u = visited.back();
15617                     visited.pop_back();
15618                     ord[u] = _n;
15619                     ids[u] = group_num;
15620                     if (u == v) break;
15621                 }
15622                 group_num++;
15623             }
15624         };
15625         for (int i = 0; i < _n; i++) {
15626             if (ord[i] == -1) dfs(dfs, i);
15627         }
15628         for (auto& x : ids) {
15629             x = group_num - 1 - x;
15630         }
15631         return {group_num, ids};
15632     }
15633
15634     std::vector<std::vector<int>> scc() {
15635         auto ids = scc_ids();
15636         int group_num = ids.first;
15637         std::vector<int> counts(group_num);
15638         for (auto x : ids.second) counts[x]++;
15639         std::vector<std::vector<int>> groups(ids.first);
15640         for (int i = 0; i < group_num; i++) {
15641             groups[i].reserve(counts[i]);
15642         }
15643         for (int i = 0; i < _n; i++) {
15644             groups[ids.second[i]].push_back(i);
15645         }
15646         return groups;
15647     }
15648
15649 private:

```

```

15650     int _n;
15651     struct edge {
15652         int to;
15653     };
15654     std::vector<std::pair<int, edge>> edges;
15655 };
15656
15657 } // namespace internal
15658
15659 } // namespace atcoder
15660
15661 namespace atcoder {
15662
15663 struct scc_graph {
15664     public:
15665         scc_graph() : internal(0) {}
15666         explicit scc_graph(int n) : internal(n) {}
15667
15668         void add_edge(int from, int to) {
15669             int n = internal.num_vertices();
15670             assert(0 <= from && from < n);
15671             assert(0 <= to && to < n);
15672             internal.add_edge(from, to);
15673         }
15674
15675         std::vector<std::vector<int>> scc() { return internal.scc(); }
15676
15677     private:
15678         internal::scc_graph internal;
15679 };
15680
15681 } // namespace atcoder
15682
15683 using namespace atcoder;
15684 using mint = modint1000000007;
15685
15686 void solve() {
15687     // code starts from here
15688     int n, m;
15689     cin >> n >> m;
15690     scc_graph graph(m);
15691
15692     mint ans = 1;
15693     mint two = 2;
15694
15695     vector<vi> adj(m);
15696     for (int u, v, i = 0; i < n; i++) {
15697         cin >> u >> v;
15698         u--;
15699         v--;
15700
15701         adj[u].pb(v);
15702         graph.add_edge(u, v);
15703     }
15704
15705     for (const vi &v : graph.scc()) {
15706         int cnt = 0;
15707         set<int> s;
15708         for (const int &i : v)
15709             s.insert(i);
15710
15711         bool sink = true;
15712
15713         for (const int &i : v)
15714             for (const int &ne : adj[i])
15715                 sink &= (s.find(ne) != s.end());
15716
15717         if (sink)
15718             ans *= two;
15719     }
15720     cout << ans << endl;
15721 }

```



```

15719         ans *= (two.pow(v.size()) - 1);
15720     else
15721         ans *= two.pow(v.size());
15722     }
15723
15724     cout << ans.val() << endl;
15725 }
15726
15727 signed main() {
15728     ios_base::sync_with_stdio(false);
15729     cin.tie(NULL);
15730     // startTime = clock();
15731
15732     int T = 1;
15733     cin >> T;
15734
15735     for (int _t = 1; _t <= T; _t++) {
15736         solve();
15737     }
15738
15739     // cerr << getCurrentTime() << endl;
15740     return 0;
15741 }
15742
15743 //MININV
15744 #ifdef WTS
15745     #include <wtsh.h>
15746 #else
15747     #include <bits/stdc++.h>
15748     using namespace std;
15749     #define dbg(...)
15750 #endif
15751
15752 #define int long long
15753 #define endl "\n"
15754 #define sz(w) (int)(w.size())
15755 using pii = pair<int, int>;
15756
15757 // ----- Input Checker Start -----
15758
15759 long long readInt(long long l, long long r, char endd)
15760 {
15761     long long x = 0;
15762     int cnt = 0, fi = -1;
15763     bool is_neg = false;
15764     while(true)
15765     {
15766         char g = getchar();
15767         if(g == '-')
15768         {
15769             assert(fi == -1);
15770             is_neg = true;
15771             continue;
15772         }
15773         if('0' <= g && g <= '9')
15774         {
15775             x *= 10;
15776             x += g - '0';
15777             if(cnt == 0)
15778                 fi = g - '0';
15779             cnt++;
15780             assert(fi != 0 || cnt == 1);
15781             assert(fi != 0 || is_neg == false);
15782             assert(!(cnt > 19 || (cnt == 19 && fi > 1)));
15783         }
15784         else if(g == endd)
15785         {
15786             if(is_neg)
15787                 x = -x;

```

```

15788         if(!(l <= x && x <= r))
15789         {
15790             cerr << "L: " << l << ", R: " << r << ", Value Found: " << x << '\n';
15791             assert(false);
15792         }
15793         return x;
15794     }
15795     else
15796     {
15797         assert(false);
15798     }
15799 }
15800 }
15801
15802 string readString(int l, int r, char endd)
15803 {
15804     string ret = "";
15805     int cnt = 0;
15806     while(true)
15807     {
15808         char g = getchar();
15809         assert(g != -1);
15810         if(g == endd)
15811             break;
15812         cnt++;
15813         ret += g;
15814     }
15815     assert(l <= cnt && cnt <= r);
15816     return ret;
15817 }
15818
15819 long long readIntSp(long long l, long long r) { return readInt(l, r, ' '); }
15820 long long readIntLn(long long l, long long r) { return readInt(l, r, '\n'); }
15821 string readStringSp(int l, int r) { return readString(l, r, ' '); }
15822 string readStringLn(int l, int r) { return readString(l, r, '\n'); }
15823 void readEOF() { assert(getchar() == EOF); }
15824
15825 vector<int> readVectorInt(int n, long long l, long long r)
15826 {
15827     vector<int> a(n);
15828     for(int i = 0; i < n - 1; i++)
15829         a[i] = readIntSp(l, r);
15830     a[n - 1] = readIntLn(l, r);
15831     return a;
15832 }
15833
15834 // ----- Input Checker End -----
15835
15836 int sumN = 0;
15837
15838 void solve()
15839 {
15840     int n = readIntLn(1, 1e5);
15841     vector<int> a = readVectorInt(n, 1, n);
15842     vector<int> pfreq(n + 2), sfreq(n + 2);
15843     for(int i = 0; i < n; i++)
15844         pfreq[a[i]]++;
15845     int ans = 0, cur = 0;
15846     for(int i = n - 1; i >= 0; i--)
15847     {
15848         // changing a[i] to a[i] + 1
15849         cur -= sfreq[a[i] - 1];
15850         cur += pfreq[a[i] + 1];
15851         ans = max(ans, cur);
15852         sfreq[a[i]]++;
15853         pfreq[a[i]]--;
15854     }
15855     cout << ans << endl;
15856 }

```

```

15857
15858 int32_t main()
15859 {
15860     ios::sync_with_stdio(0);
15861     cin.tie(0);
15862     int T = readIntLn(1, 1e5);
15863     for(int tc = 1; tc <= T; tc++)
15864     {
15865         // cout << "Case #" << tc << ": ";
15866         solve();
15867     }
15868     assert(sumN <= 2e5);
15869     readEOF();
15870     return 0;
15871 }
15872
15873 //POWTREE
15874 #include <bits/stdc++.h>
15875 using namespace std;
15876
15877 #define ll long long
15878 #define vi vector<int>
15879 #define pb push_back
15880 #define allrev(v) v.rbegin(), v.rend()
15881 #define lb lower_bound
15882 const int mod = 1e9+7;
15883 const int N = 1e5+5;
15884
15885 vi g[N];
15886 void dfs(int u,int p,vi &ass, vi&a)
15887 {
15888     ass[u] = a[u];
15889     for(int v:g[u])
15890     {
15891         if(v!=p)
15892         {
15893             dfs(v,u,ass,a);
15894             ass[u] = max(ass[u],ass[v]);
15895         }
15896     }
15897 }
15898
15899 void dfs2(int u,int p,vi &ass,vector<ll> &b, vector<ll> &cur,vector<ll> &pref,ll rem)
15900 {
15901     cur.pb(ass[u]);
15902     pref.pb(ass[u]);
15903     pref.back() += pref[pref.size()-2];
15904     ll l = ass[u],r = 1e12;
15905     while(l+1<r)
15906     {
15907         ll mid = (l+r)/2;
15908         ll ind = lb(allrev(cur),mid)-cur.rbegin();
15909         ll sc = ind*mid-(pref[pref.size()-1]-pref[pref.size()-1-ind]);
15910         if(sc>=rem)r = mid;
15911         else l = mid;
15912     }
15913     b[u] = r;
15914     for(int v:g[u])
15915     {
15916         if(v!=p)
15917         {
15918             dfs2(v,u,ass,b,cur,pref,rem);
15919             ass[u] = max(ass[u],ass[v]);
15920         }
15921     }
15922     cur.pop_back();
15923     pref.pop_back();
15924 }
15925

```

```

15926 void solve()
15927 {
15928     ll n,x;
15929     cin >> n >> x;
15930     for(int i = 0; i<=n; i++)g[i].clear();
15931     vi a(n+1);
15932
15933     for(int i = 1; i<=n; i++)cin >> a[i];
15934
15935     for(int i = 1; i<n; i++)
15936     {
15937         int u,v;
15938         cin >> u >> v;
15939         g[u].pb(v);
15940         g[v].pb(u);
15941     }
15942     vi ass(n+1,0);
15943     dfs(1,-1,ass,a);
15944     ll sum = 0;
15945     for(int i:ass)sum += i;
15946     if(sum>=x)
15947     {
15948         cout << "0\n";
15949         return;
15950     }
15951     vector<ll> b(n+1,-1);
15952     vector<ll> cur,pref;
15953     pref.pb(0);
15954     dfs2(1,-1,ass,b,cur,pref,-sum+x);
15955     ll ans = 1e12;
15956     for(int i = 1; i<=n; i++)
15957     {
15958         ans = min(ans,b[i]-a[i]);
15959         assert(b[i]!=-1);
15960     }
15961     cout << ans << "\n";
15962 }
15963
15964
15965 int main()
15966 {
15967     ios_base::sync_with_stdio(0); cin.tie(0);cout.tie(0);
15968     int t = 1;
15969     cin>> t;
15970     for(int i = 1; i<=t; i++) {
15971         solve();
15972
15973
15974
15975     }
15976     return 0;
15977 }
15978
15979 //MINORPATH
15980
15981 #include<bits/stdc++.h>
15982
15983 #include<ext/pb_ds/assoc_container.hpp>
15984 #include<ext/pb_ds/tree_policy.hpp>
15985
15986 using namespace __gnu_pbds;
15987 using namespace std;
15988 #define ff first
15989 #define ss second
15990 #define infinity 8999999999999999999
15991 #define sz(v) ((int)(v).size())
15992 #define all(v) (v).begin(),(v).end()
15993 #define MOD_DEFINE const int MOD = 1e9 + 7;
15994 #define endl '\n'

```

```

15995     #define int                long long
15996     #define pii               pair<int, int>
15997     #define vi                vector<int>
15998     #define pb(n)             push_back((n))
15999     #define mii               map<int, int>
16000     #define umii              unordered_map<int, int>
16001     #define l(var, initial, final) for(int var=initial; var < final; var++)
16002     #define cout              std::cout
16003     #define cin               std::cin
16004     #define pqb               priority_queue<int>
16005     #define pqs               priority_queue<int, vi, greater<int>>
16006     #define fps(x, y)         fixed<<setprecision(y)<<x
16007     typedef long long ll;
16008     typedef vector<pii> vpii;
16009     typedef tree<int, null_type, less<int>, rb_tree_tag, tree_order_statistics_node_update>
pbds;

16010
16011     void prn() { }
16012     template<typename T1, typename T2> istream &operator >> (istream& in, pair<T1, T2> &a){
in >> a.ff >> a.ss; return in;}
16013     template<typename T1, typename T2> ostream &operator << (ostream& out, pair<T1, T2> a){
out << a.ff << ' ' << a.ss; return out;}
16014     template<typename T, typename T1> T amax(T &a, T1 b){if(b > a) a = b; return a;}
16015     template<typename T, typename T1> T amin(T &a, T1 b){if(b < a) a = b; return a;}
16016     template<typename T> istream& operator>>(istream &in, vector<T> &v) { for (auto &x : v)
in >> x; return in;}
16017     template<typename T> ostream& operator<<(ostream &out, vector<T> &v) {out << "{ "; for (
auto &x : v) out << x << " "; out << "}\n"; return out;}
16018     template<typename T, typename... Args> void prn(T x, Args... args) {cout << x << " ";
prn(args...);}
16019     template<typename Iterable> void prnIter(const Iterable& ITER, ostream&out = cout){ auto
x = ITER.begin(); out << "{ "; for (; x != ITER.end(); ++x) out << *x << ' '; out <<
"}" << endl;}

16020
16021     MOD_DEFINE
16022
16023     bool traverse(int bit, const vector<bool> &safe, const vector<int> &in){
16024         int n = in.size();
16025
16026         if(((in[0] >> bit) & 1) or ((in.back() >> bit) & 1)){
16027             return false;
16028         }
16029
16030         int R = in[0];
16031
16032         int mx = 0;
16033         for(int cur = 0; R < n - 1; cur = R + 1, R = mx){
16034
16035             for(int i = cur; i <= R; i++){
16036                 if(((in[i] >> bit) & 1) or !safe[i]) continue;
16037
16038                 amax(mx, i + in[i]);
16039             }
16040             if(mx <= R){
16041                 return false;
16042             }
16043         }
16044         return true;
16045     }
16046     void slv(){
16047         int n; cin >> n;
16048         vector<int> in(n); cin >> in;
16049
16050         vector<bool> safe(n, true);
16051         vector<bool> f(32, true);
16052
16053         if(!traverse(31, safe, in)){
16054             cout << -1 << endl; return;
16055         }

```

```

16056         for(int bit = 31; bit >= 0; bit--){
16057             // try to keep bit off
16058             if(traverse(bit, safe, in)){
16059                 f[bit] = 0;
16060
16061                 for(int i = 0; i < n; i++){
16062                     if((in[i] >> bit) & 1) safe[i] = false;
16063                 }
16064             }
16065         }
16066         int ans = 0;
16067         for(int i = 0; i < 32; i++){
16068             if(f[i])
16069                 ans += (1 << i);
16070         }
16071         cout << ans << endl;
16072     }
16073
16074     int32_t main(){
16075
16076         ios_base::sync_with_stdio(false); cin.tie(NULL); cout.tie(NULL);
16077
16078
16079         int T = 1;
16080
16081         cin >> T;
16082         for(int t = 1; t <= T; t++){
16083             // cout << "Case #" << T << ": ";
16084             slv();
16085         }
16086         return 0;
16087     }
16088     /*
16089     *think brute force first.
16090     *try proving the algorithm on pen n paper first.
16091     *floating point precision errors ?
16092     *implementation too lengthy ? logic might be incorrect.
16093     *read the question again.
16094     */
16095
16096     //FTOL
16097     #include"bits/stdc++.h"
16098     using namespace std;
16099     #define PB push_back
16100     #define ll long long
16101
16102     #ifdef LOCAL
16103     #include"bits/debug.h"
16104     #else
16105     #define dbg(...) 0
16106     #endif
16107
16108     #define I ios::sync_with_stdio(false); cin.tie(0);
16109     #define Q int tt; cin>>tt ; for(int qq=1; qq <= tt; qq++)
16110
16111     using pii = pair<int, int>;
16112     using pll = pair<ll, ll>;
16113     #define x first
16114     #define y second
16115
16116
16117
16118     int main() {
16119         I
16120         Q {
16121             int n, m, k;
16122             cin >> n >> m >> k;
16123             vector<pii> a(k);
16124             for (int i = 0; i < k; i++) {

```

```

16125         cin >> a[i].x >> a[i].y;
16126     }
16127     sort(a.begin(), a.end());
16128
16129     vector<int> lis(k + 1, 1e9);
16130     lis[0] = -1e9;
16131     int cut = 0;
16132     for (int i = 0; i < k;) {
16133         vector<pii> tmp;
16134         int il = i;
16135         while(i < k && a[i].x == a[il].x) {
16136             if(a[i].x != n && a[i].y != m) {
16137                 int j = upper_bound(lis.begin(), lis.end(), a[i].y) - lis.begin();
16138                 if(lis[j - 1] < a[i].y) {
16139                     tmp.pb({j, a[i].y});
16140                     cut = max(cut, j);
16141                 }
16142             }
16143             i++;
16144         }
16145         for(auto [id, y] : tmp) {
16146             lis[id] = min(lis[id], y);
16147         }
16148     }
16149     cout << (n + m - 2 - cut) << "\n";
16150 }
16151
16152 return 0;
16153 }
16154
16155 //HAMTREE
16156 #pragma GCC optimize("O3,unroll-loops")
16157 #pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
16158
16159 #include <bits/stdc++.h>
16160 using namespace std;
16161
16162 // #include <ext/pb_ds/assoc_container.hpp> //required
16163 // #include <ext/pb_ds/tree_policy.hpp> //required
16164 // using namespace __gnu_pbds; //required
16165 // template <typename T> using ordered_set = tree<T, null_type, less<T>,
16166 // rb_tree_tag, tree_order_statistics_node_update>;
16167
16168 // ordered_set <int> s;
16169 // s.find_by_order(k); returns the (k+1)th smallest element
16170 // s.order_of_key(k); returns the number of elements in s strictly less than k
16171
16172 #define pb push_back
16173 #define mp(x, y) make_pair(x, y)
16174 #define all(x) x.begin(), x.end()
16175 #define allr(x) x.rbegin(), x.rend()
16176 #define leftmost_bit(x) (63 - __builtin_clzll(x))
16177 #define rightmost_bit(x) __builtin_ctzll(x) // count trailing zeros
16178 #define set_bits(x) __builtin_popcountll(x)
16179 #define pow2(i) (1LL << (i))
16180 #define is_on(x, i) ((x)&pow2(i)) // state of the ith bit in x
16181 #define set_on(x, i) ((x) | pow2(i)) // returns integer x with ith bit on
16182 #define set_off(x, i) ((x) & ~pow2(i)) // returns integer x with ith bit off
16183 #define fi first
16184 #define se second
16185
16186 typedef long long int ll;
16187 typedef long double ld;
16188
16189 const int MOD = 1e9 + 7; // 998244353;
16190 const int MX = 2e5 + 5;
16191 const int INF = 1e9; // not too close to LLONG_MAX
16192 const ld PI = acos((ld)-1);
16193 const ld EPS = 1e-8;

```

```

16194     const int dx[4] = {1, 0, -1, 0},
16195             dy[4] = {0, 1, 0, -1}; // for every grid problem!!
16196
16197 // hash map and operator overload from
16198 // https://www.youtube.com/watch?v=jkfa0Ts6YBA Custom hash map
16199 struct custom_hash {
16200     static uint64_t splitmix64(uint64_t x) {
16201         x += 0x9e3779b97f4a7c15;
16202         x = (x ^ (x >> 30)) * 0xbf58476d1ce4e5b9;
16203         x = (x ^ (x >> 27)) * 0x94d049bb133111eb;
16204         return x ^ (x >> 31);
16205     }
16206
16207     size_t operator()(uint64_t x) const {
16208         static const uint64_t FIXED_RANDOM =
16209             chrono::steady_clock::now().time_since_epoch().count();
16210         return splitmix64(x + FIXED_RANDOM);
16211     }
16212 };
16213 template <typename T1, typename T2> // Key should be integer type
16214 using safe_map = unordered_map<T1, T2, custom_hash>;
16215
16216 // Operator overloads
16217 template <typename T1, typename T2> // cin >> pair<T1, T2>
16218 istream &operator>>(istream &istream, pair<T1, T2> &p) {
16219     return (istream >> p.first >> p.second);
16220 }
16221 template <typename T1, typename T2> // cout << pair<T1, T2>
16222 ostream &operator<<(ostream &ostream, const pair<T1, T2> &p) {
16223     return (ostream << p.first << " " << p.second);
16224 }
16225
16226 template <typename T> // cin >> array<T, 2>
16227 istream &operator>>(istream &istream, array<T, 2> &p) {
16228     return (istream >> p[0] >> p[1]);
16229 }
16230 template <typename T> // cout << array<T, 2>
16231 ostream &operator<<(ostream &ostream, const array<T, 2> &p) {
16232     return (ostream << p[0] << " " << p[1]);
16233 }
16234
16235 template <typename T> // cin >> vector<T>
16236 istream &operator>>(istream &istream, vector<T> &v) {
16237     for (auto &it : v)
16238         cin >> it;
16239     return istream;
16240 }
16241 template <typename T> // cout << vector<T>
16242 ostream &operator<<(ostream &ostream, const vector<T> &c) {
16243     for (auto &it : c)
16244         cout << it << " ";
16245     return ostream;
16246 }
16247 clock_t startTime;
16248 mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());
16249 double getCurrentTime() {
16250     return (double)(clock() - startTime) / CLOCKS_PER_SEC;
16251 }
16252 string to_string(string s) { return "'" + s + "'"; }
16253 string to_string(const char *s) { return to_string((string)s); }
16254 string to_string(bool b) { return (b ? "true" : "false"); }
16255 int getRandomNumber(int l, int r) {
16256     uniform_int_distribution<int> dist(l, r);
16257     return dist(rng);
16258 }
16259
16260 // https://github.com/the-tourist/algo/blob/master/misc/debug.cpp
16261 template <typename A, typename B> string to_string(pair<A, B> p) {
16262     return "(" + to_string(p.first) + ", " + to_string(p.second) + ")";

```



```

16263 }
16264 template <typename A> string to_string(A v) {
16265     bool first = true;
16266     string res = "{";
16267     for (const auto &x : v) {
16268         if (!first) {
16269             res += ", ";
16270         }
16271         first = false;
16272         res += to_string(x);
16273     }
16274     res += "}";
16275     return res;
16276 }
16277 void debug_out() { cerr << endl; }
16278 template <typename Head, typename... Tail> void debug_out(Head H, Tail... T) {
16279     cerr << " " << to_string(H);
16280     debug_out(T...);
16281 }
16282
16283 #ifdef LOCAL_DEBUG
16284 #define debug(...) cerr << "[" << #__VA_ARGS__ << "]:", debug_out(__VA_ARGS__)
16285 #else
16286 #define debug(...) ;
16287 #endif
16288
16289 #define int ll // disable when you want to use atcoder library
16290 #define endl '\n' // disable when dealing with interactive problems
16291
16292 typedef vector<int> vi;
16293 typedef pair<int, int> pii;
16294 typedef array<int, 2>
16295     edge; // for graphs, make it array<int,3> for weighted edges
16296
16297 // #include <atcoder/all>
16298 // using namespace atcoder;
16299
16300 constexpr int MAXN = 2e5;
16301 vector<vi> adj;
16302 int root = 0;
16303 vector<array<int, 2>> dp;
16304 // dp[i][0] = minimum number of edges to remove in the subtree rooted at i such
16305 // that the resulting graph is a forest of paths && the edge between i and its
16306 // parent is removed
16307
16308 // dp[i][1] = minimum number of edges to remove in the subtree rooted at i such
16309 // that the resulting graph is a forest of paths && the edge between i and its
16310 // parent is not removed
16311
16312 // dp[root][1] = infinity
16313 // Final answer = dp[root][0]
16314 void dfs(int cur, int par) {
16315     // degree(cur) in the final graph is <= 2.
16316     // It is not optimal for degree to be 0. So degree(cur) = 0 or 1 or 2
16317     // in the state dp[cur][1], cur can have atmost one child.
16318     // in the state dp[cur][0], cur can have atmost two children.
16319
16320     vi vec;
16321     int sum = 0;
16322     for (int ne : adj[cur]) {
16323         if (ne == par)
16324             continue;
16325
16326         dfs(ne, cur);
16327         sum += dp[ne][0];
16328         vec.pb(dp[ne][0] - dp[ne][1]);
16329     }
16330     int d = vec.size();
16331

```

```

16332     if (d == 0) { // leaf
16333         dp[cur][0] = dp[cur][1] = 0;
16334         return;
16335     }
16336
16337     sort(all(vec));
16338     reverse(all(vec));
16339
16340     dp[cur][0] = dp[cur][1] = d - 1 + sum - vec[0];
16341
16342     if (vec.size() > 1)
16343         dp[cur][0] = min(dp[cur][0], d - 2 + sum - vec[0] - vec[1]);
16344 }
16345
16346 void solve() {
16347     // code starts from here
16348     int N;
16349     cin >> N;
16350     adj.clear();
16351     adj.resize(N);
16352     dp.assign(N, {INF, INF});
16353
16354     for (int u, v, i = 0; i < N - 1; i++) {
16355         cin >> u >> v;
16356         u--;
16357         v--;
16358         adj[u].pb(v);
16359         adj[v].pb(u);
16360     }
16361
16362     dfs(0, 0);
16363     cout << dp[0][0] << endl;
16364 }
16365
16366 signed main() {
16367     ios_base::sync_with_stdio(false);
16368     cin.tie(NULL);
16369     // startTime = clock();
16370
16371     int T = 1;
16372     cin >> T;
16373
16374     for (int _t = 1; _t <= T; _t++) {
16375         solve();
16376     }
16377
16378     // cerr << getCurrentTime() << endl;
16379     return 0;
16380 }
16381
16382 //ELEVATORS
16383 #include <bits/stdc++.h>
16384 using namespace std;
16385
16386 typedef long long ll;
16387
16388 ll T, N, M, H, A[500005], B[500005];
16389
16390 bool test(ll x)
16391 {
16392     ll cur_id = 0;
16393
16394     for (ll i = 0; i < M; i++)
16395     {
16396         ll P = (x - B[cur_id] + 1) / (2 * H);
16397         cur_id += P;
16398
16399         if (P <= 0)
16400         {

```

```

16401         return false;
16402     }
16403
16404     if (cur_id >= N)
16405     {
16406         return true;
16407     }
16408 }
16409
16410 return false;
16411 }
16412
16413 void solve()
16414 {
16415     cin >> N >> M >> H;
16416
16417     for (ll i = 0; i < N; i++)
16418     {
16419         cin >> A[i] >> B[i];
16420     }
16421
16422     sort(B, B + N, greater<ll>());
16423
16424     // binary search
16425     ll lower = 0, upper = (ll)1e18, ans = (ll)1e18;
16426
16427     while (lower <= upper)
16428     {
16429         ll mid = (lower + upper) / 2;
16430
16431         if (test(mid))
16432         {
16433             ans = min(ans, mid);
16434             upper = mid - 1;
16435         }
16436         else
16437         {
16438             lower = mid + 1;
16439         }
16440     }
16441
16442     cout << ans << "\n";
16443 }
16444
16445 int main()
16446 {
16447     ios_base::sync_with_stdio(0);
16448     cin.tie(0);
16449
16450     cin >> T;
16451
16452     for (ll t = 0; t < T; t++)
16453     {
16454         solve();
16455     }
16456
16457     return 0;
16458 }
16459
16460 //MAXGRAPH
16461 #include<bits/stdc++.h>
16462 #define still_me main
16463 #define endl "\n"
16464 #define int long long int
16465 #define all(a) (a).begin() , (a).end()
16466 #define print(a) for(auto TEMPORARY: a) cerr<<TEMPORARY<<" ";cerr<<endl;
16467 #define tt int TESTCASE;cin>>TESTCASE;while(TESTCASE--)
16468 #define arrin(a,n) for(int INPUT=0;INPUT<n;INPUT++)cin>>a[INPUT]
16469

```

```

16470 using namespace std;
16471 const int mod = 1e9+7;
16472 const int inf = 1e18;
16473 int N = 0;
16474
16475 pair<int,int> ed(int n , int m) {
16476     int a = -1 , b = -1;
16477     int ans = 0;
16478     for(int i=(m+n-1)/n;i<m;i++) {
16479         int A = i;
16480         int B = n - (A*n+m-1) / m;
16481         if(B == 0)
16482             continue;
16483         if((A*n + B*m) > ans) {
16484             ans = (A*n + B*m);
16485             a = A;
16486             b = B;
16487         }
16488     }
16489     return {a , b};
16490 }
16491
16492 void solve() {
16493     int n;
16494     cin>>n;
16495     N+=n;
16496     vector<int> a(n);
16497     arrin(a , n);
16498
16499     int k = *max_element(all(a)) + 1;
16500     vector<vector<int>> c(k);
16501
16502     for(int i=0;i<n;i++){
16503         if(a[i] >= k) {
16504             print(a);
16505             cerr<<n<<" "<<a[i]<<endl;
16506             break;
16507         }
16508         c[a[i]].push_back(i);
16509     }
16510     // print(c[0]);
16511     // print(c[1]);
16512
16513     if(k != 2) {
16514         vector<vector<int>> adj(n);
16515         for(int i=0;i<3;i++) {
16516             for(int j: c[i]) {
16517                 for(int l: c[(i+1)%3])
16518                     adj[j].push_back(l);
16519             }
16520         }
16521         for(int i=3;i<k;i++) {
16522             for(int j: c[i]){
16523                 for(int l: c[0])
16524                     adj[j].push_back(l);
16525             }
16526             for(int j=1;j<i;j++) {
16527                 for(int l: c[i]) {
16528                     for(int m: c[j]) {
16529                         adj[m].push_back(l);
16530                     }
16531                 }
16532             }
16533         }
16534         int ans = 0;
16535         for(auto i: adj) {
16536             ans += i.size();
16537         }
16538         cout<<ans<<endl;

```

```

16539     for(int i=0;i<n;i++) {
16540         for(auto j: adj[i])
16541             cout<<i+1<<" "<<j+1<<endl;
16542     }
16543     // cout<<"done\n";
16544     return;
16545 }
16546 if(c[0].size() > c[1].size())
16547     swap(c[0] , c[1]);
16548 // print(c[0]);
16549 // print(c[1]);
16550 auto x = ed(c[0].size() , c[1].size());
16551
16552 if(x.first == -1) {
16553     cout<<-1<<endl;
16554     return;
16555 }
16556
16557 int l = c[0].size() , r = c[1].size();
16558 // cerr<<l<<" "<<r<<endl;
16559 // if(l+r != n){
16560 //     cerr<<"uneq\n";
16561 //     return;
16562 // }
16563 vector<vector<int>> adj(n);
16564 vector<set<int>> fout(l) , sout(r) , fin(l) , sin(r);
16565 set<pair<int,int>> sindeg , findeg;
16566 for(int i=0;i<l;i++)
16567     findeg.insert({0 , i});
16568 for(int j=0;j<r;j++)
16569     sindeg.insert({(0) , j});
16570 // for(int i=0;i<l;i++) {
16571 //     adj[c[0][i]].push_back(c[1][i]);
16572 //     adj[c[1][i]].push_back(c[0][(i+1) % l]);
16573 // }
16574 //     fout[i].insert(i);
16575 //     sout[i].insert((i+1) % l);
16576 //     fin[(i+1) % l].insert(i);
16577 //     sin[i].insert(i);
16578 // }
16579
16580 for(int i=r-1;i>=0;i--) {
16581     set<pair<int,int>> used;
16582     for(auto j: findeg) {
16583         if(used.size()+adj[c[1][i]].size() == x.second)
16584             break;
16585         if(sout[i].count(j.second) || sin[i].count(j.second))
16586             continue;
16587         used.insert(j);
16588     }
16589     if(used.size()+adj[c[1][i]].size() != x.second) {
16590         cerr << "here2\n";
16591         cerr<<x.first<<" "<<x.second<<endl;
16592         cerr<<l<<" "<<r<<endl;
16593         cerr<<used.size()<<" "<<adj[c[1][i]].size()<<" "<<x.second<<endl;
16594         cerr<<sin[i].size()<<endl;
16595         // return;
16596     }
16597     for(auto j: used) {
16598         sout[i].insert(j.second);
16599         fin[j.second].insert(i);
16600         adj[c[1][i]].push_back(c[0][j.second]);
16601         findeg.erase(j);
16602         findeg.insert({j.first+1 , j.second});
16603     }
16604 }
16605
16606 for(int i=0;i<l;i++) {

```

```

16608     set<pair<int,int>> used;
16609     for(auto j: sindeg) {
16610         if(used.size()+adj[c[0][i]].size() == x.first)
16611             break;
16612         if(fout[i].count(j.second) || fin[i].count(j.second))
16613             continue;
16614         used.insert(j);
16615     }
16616     if(used.size()+adj[c[0][i]].size() != x.first) {
16617         cerr << "here1\n";
16618     }
16619     for(auto j: used) {
16620         // cerr<<j.first<<" "<<j.second<<endl;
16621         fout[i].insert(j.second);
16622         sin[j.second].insert(i);
16623         adj[c[0][i]].push_back(c[1][j.second]);
16624         sindeg.erase(j);
16625         sindeg.insert({j.first+1 , j.second});
16626     }
16627     // cerr<<endl;
16628 }
16629 // if(sindeg.count({2 , 4})) {
16630 //     cerr<<"YES\n";
16631 // }
16632
16633
16634 int kk = 0;
16635 cout<<l*x.first + r*x.second<<endl;
16636 for(int i=0;i<n;i++) {
16637     for(int j: adj[i]) {
16638         cout<<i+1<<" "<<j+1<<endl;
16639         kk++;
16640     }
16641 }
16642 if(l*x.first + r*x.second != kk) {
16643     cerr<<l<<" "<<r<<" "<<kk <<" "<< l*x.first + r*x.second<<endl;
16644 }
16645 }
16646
16647 signed still_me()
16648 {
16649     ios_base::sync_with_stdio(false);cin.tie(NULL);cout.tie(NULL);
16650     // freopen("6.in" , "r" , stdin);
16651     // freopen("6.out" , "w" , stdout);
16652     tt{
16653         solve();
16654     }
16655     // assert(N <= 1000);
16656     return 0;
16657 }
16658
16659 //NOSEQ-HORNER ALG
16660 // Code by Sahil Tiwari (still_me)
16661
16662 #include<bits/stdc++.h>
16663 #define still_me main
16664 #define endl "\n"
16665 #define int long long int
16666 #define all(a) (a).begin() , (a).end()
16667 #define print(a) for(auto TEMPORARY: a) cout<<TEMPORARY<<" ";cout<<endl;
16668 #define tt int TESTCASE;cin>>TESTCASE;while(TESTCASE--)
16669 #define arrin(a,n) for(int INPUT=0;INPUT<n;INPUT++)cin>>a[INPUT]
16670
16671 using namespace std;
16672 const int mod = 1e18;
16673 const int inf = 1e18;
16674
16675 long long power(long long a , long long b , long long mod){
16676     if(b==0)

```

```

16677         return 1;
16678     long long res = power(a , b/2 , mod);
16679     res = res*res % mod;
16680     if(b%2)
16681         res = res*a % mod;
16682     return res;
16683 }
16684 int t = 0;
16685
16686 void solve() {
16687     t++;
16688     int n , k , s;
16689     cin>>n>>k>>s;
16690     vector<int> b(n);
16691     int i = 0;
16692     bool flag = 1;
16693     while(i < min(n , 6111)) {
16694         if(s % k == 0) {
16695             s /= k;
16696         }
16697         else if((s-1) % k == 0) {
16698             s = (s-111) / k;
16699             b[i] = 1;
16700         }
16701         else if((s+1) % k == 0) {
16702             s = (s+111) / k;
16703             b[i] = -1;
16704         }
16705         else {
16706             flag = 0;
16707             // assert(false);
16708             // cerr<<t<<endl;
16709             break;
16710         }
16711         i++;
16712     }
16713     // cout<<s<<endl;
16714     if(!flag || s != 0) {
16715         cout<<"-2"<<endl;
16716     }
16717     else {
16718         print(b);
16719     }
16720 }
16721
16722 signed still_me()
16723 {
16724     ios_base::sync_with_stdio(false);cin.tie(NULL);cout.tie(NULL);
16725     // freopen("3.in" , "r" , stdin);
16726     // freopen("3.out" , "w" , stdout);
16727     tt{
16728         solve();
16729     }
16730     cerr << "time taken : " << (float)clock() / CLOCKS_PER_SEC << " secs" << endl;
16731     return 0;
16732 }
16733
16734 //GSEQ
16735 #include "bits/stdc++.h"
16736 // #pragma GCC optimize("O3,unroll-loops")
16737 // #pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
16738 using namespace std;
16739 using ll = long long int;
16740 mt19937_64 rng(chrono::high_resolution_clock::now().time_since_epoch().count());
16741
16742 int main()
16743 {
16744     ios::sync_with_stdio(false); cin.tie(0);
16745

```

```

16746
16747     int t; cin >> t;
16748     while (t--) {
16749         int n; cin >> n;
16750         vector<int> a(n);
16751         for (int i = 0; i < n; ++i) cin >> a[i];
16752         vector<int> difs = {0};
16753         for (int i = 0; i < n; ++i) {
16754             if (a[i] == 1) difs.push_back(difs[i] + 1);
16755             else difs.push_back(difs[i] - 1);
16756         }
16757
16758         map<int, int> len, last;
16759         vector<int> link(n+1, -1), ending_at(n+1);
16760         for (int i = 0; i <= n; ++i) {
16761             int cur = len[difs[i]-1] + 1;
16762             ending_at[i] = cur;
16763             if (cur > 1) link[i] = last[difs[i] - 1];
16764
16765             len[difs[i]] = cur;
16766             last[difs[i]] = i;
16767         }
16768         int ans = *max_element(begin(ending_at), end(ending_at));
16769         cout << ans << '\n';
16770         for (int i = n; i >= 0; --i) {
16771             if (ending_at[i] != ans) continue;
16772             vector<int> pos;
16773             int cur = i;
16774             while (1) {
16775                 pos.push_back(cur);
16776                 if (link[cur] == -1) break;
16777                 cur = link[cur];
16778             }
16779             reverse(begin(pos), end(pos));
16780             for (int x : pos) cout << x+1 << ' ';
16781             cout << '\n';
16782             break;
16783         }
16784     }
16785 }
16786
16787 //GGTREE
16788 // Code by Sahil Tiwari (still_me)
16789
16790 #include<bits/stdc++.h>
16791 #define still_me main
16792 #define endl "\n"
16793 #define int long long int
16794 #define all(a) (a).begin() , (a).end()
16795 #define print(a) for(auto TEMPORARY: a) cout<<TEMPORARY<<" ";cout<<endl;
16796 #define tt int TESTCASE;cin>>TESTCASE;while(TESTCASE--)
16797 #define arrin(a,n) for(int INPUT=0;INPUT<n;INPUT++)cin>>a[INPUT]
16798
16799 using namespace std;
16800 const int mod = 1e9+7;
16801 const int inf = 1e18;
16802
16803 long long power(long long a , long long b , long long mod){
16804     if(b==0)
16805         return 1;
16806     long long res = power(a , b/2 , mod);
16807     res = res*res%mod;
16808     if(b%2)
16809         res = res*a % mod;
16810     return res;
16811 }
16812
16813 int inverse(int a){
16814     return power(a , mod-2 , mod);

```



```

16815     }
16816     map<int,int> p;
16817     int cnt = 0;
16818     void dfs(vector<vector<int>> &adj , int j , int prev , int prob) {
16819         if(adj[j].size() == 1 && j != 0) {
16820             p[j] = prob;
16821             return;
16822         }
16823         for(int &i: adj[j]) {
16824             if(i == prev)
16825                 continue;
16826             dfs(adj , i , j , prob * inverse(adj[j].size() - (j == 0 ? 0 : 1)) % mod);
16827         }
16828     }
16829     int ans = 0;
16830     struct Trie{
16831         vector<array<int, 2>> node;
16832         vector<int> last;
16833         vector<pair<int, int>> bck;
16834         Trie() {
16835             node.push_back({-1, -1});
16836             last.push_back(-1);
16837             bck.push_back({-1, -1});
16838         }
16839         void insert(int val, int n) {
16840             int cur = 0;
16841             for(int i = 29 ; i >= 0 ; i--) {
16842                 int p = (val >> i) & 1;
16843                 if(node[cur][p] == -1) {
16844                     node[cur][p] = node.size();
16845                     node.push_back({-1, -1});
16846                     last.push_back(n);
16847                     bck.push_back({cur, p});
16848                 }
16849                 cur = node[cur][p];
16850             }
16851         }
16852         void Delete(int n) {
16853             while(last.back() == n) {
16854                 node[bck.back().first][bck.back().second] = -1;
16855                 bck.pop_back();
16856                 last.pop_back(), node.pop_back();
16857             }
16858         }
16859         int query(int v) {
16860             int cur = 0, ans = 0;
16861             for(int i = 29 ; i >= 0 ; i--) {
16862                 int p = (v >> i) & 1;
16863                 if(node[cur][1 ^ p] > 0)
16864                     ans ^= 1 << i, cur = node[cur][1 ^ p];
16865                 else cur = node[cur][p];
16866             }
16867             return ans;
16868         }
16869     };
16870
16871     void tdfs(vector<vector<int>> &adj , vector<int> &a, int j , int prev, int curr, Trie &T
16872 ) {
16873         T.insert(a[j] , j);
16874         curr ^= a[j];
16875         // cout<<curr<<endl;
16876         if(adj[j].size() == 1 && j != 0) {
16877             ans += p[j] * T.query(curr);
16878             ans %= mod;
16879         }
16880         for(int &i: adj[j]) {
16881             if(i == prev)
16882                 continue;
16883             tdfs(adj , a , i , j , curr, T);

```

```

16883     }
16884     T.Delete(j);
16885 }
16886
16887 void chal_bsdk() {
16888     p.clear();
16889     ans = 0;
16890     int n;
16891     cin>>n;
16892     vector<int> a(n);
16893     arrin(a , n);
16894     vector<vector<int>> adj(n);
16895     for(int i=0;i<n-1;i++) {
16896         int u , v;
16897         cin>>u>>v;
16898         u--;v--;
16899         adj[u].push_back(v);
16900         adj[v].push_back(u);
16901     }
16902
16903     dfs(adj , 0 , 0 , 1);
16904     Trie T;
16905     tdfs(adj , a , 0 , 0 , 0 , T);
16906     cout<<ans<<endl;
16907
16908 }
16909
16910 signed still_me()
16911 {
16912     ios_base::sync_with_stdio(false);cin.tie(NULL);cout.tie(NULL);
16913
16914     // freopen("15.in" , "r" , stdin);
16915     // freopen("15.out" , "w" , stdout);
16916     tt{
16917         chal_bsdk();
16918     }
16919     return 0;
16920 }
16921
16922 //GARRANGE
16923 // Code by Sahil Tiwari (still_me)
16924
16925 #include<bits/stdc++.h>
16926 #define still_me main
16927 #define endl "\n"
16928 #define int long long int
16929 #define all(a) (a).begin() , (a).end()
16930 #define print(a) for(auto TEMPORARY: a) cout<<TEMPORARY<<" ";cout<<endl;
16931 #define tt int TESTCASE;cin>>TESTCASE;while(TESTCASE--)
16932 #define arrin(a,n) for(int INPUT=0;INPUT<n;INPUT++)cin>>a[INPUT]
16933
16934 using namespace std;
16935 const int mod = 1e9+7;
16936 const int inf = 1e18;
16937
16938 const int N = 1e6;
16939 int fact[N+1];
16940 void factorial(){
16941     fact[0] = fact[1] = 1;
16942     for(int i=2;i<=N;i++){
16943         fact[i] = (fact[i-1] * i) % mod;
16944     }
16945 }
16946
16947 long long power(long long a , long long b , long long mod){
16948     if(b==0)
16949         return 1;
16950     long long res = power(a , b/2 , mod);
16951     res = res*res%mod;

```

```

16952     if(b%2)
16953         res = res*a % mod;
16954     return res;
16955 }
16956
16957 int inverse(int a){
16958     return power(a , mod-2 , mod);
16959 }
16960
16961 int nCr(int n , int r){
16962     if(r>n)
16963         return 0;
16964     if(r < 0)
16965         return 0;
16966     return fact[n] * (inverse(fact[r]) * inverse(fact[n-r]) % mod) % mod;
16967 }
16968
16969
16970 void chal_bsdk() {
16971     int n;
16972     cin>>n;
16973     vector<int> a(n);
16974     arrin(a , n);
16975     sort(all(a));
16976
16977     int l = 0;
16978     int ans = power(2 , n-1 , mod);
16979     while(l < n) {
16980         int r = upper_bound(all(a) , a[l]) - a.begin() - 1;
16981         // cout<<r<<endl;
16982         if(r == l) {
16983             l++;
16984             continue;
16985         }
16986         {
16987             int k = l + n - r - 1;
16988             int x = (nCr(k , l) * (power(2 , r-1 , mod) - 1 + mod)) % mod;
16989             ans = (ans - x + mod) % mod;
16990         }
16991         for(int i=2;i<(r-l+1);i++) {
16992             int L = l;
16993             int R = n-r-1;
16994             int g = L+R + (r-l-i);
16995
16996             // Left side
16997             int x = nCr(g , L-1) * (power(2 , i-1 , mod) - 1 + mod) % mod;
16998             ans = (ans - x + mod) % mod;
16999
17000             // Right side
17001             x = nCr(g , R-1) * (power(2 , i-1 , mod) - 1 + mod) % mod;
17002             ans = (ans - x + mod) % mod;
17003         }
17004         l = r+1;
17005     }
17006     cout<<ans<<endl;
17007 }
17008
17009 signed still_me()
17010 {
17011     ios_base::sync_with_stdio(false);cin.tie(NULL);cout.tie(NULL);
17012
17013     // freopen("4.in" , "r" , stdin);
17014     // freopen("4.out" , "w" , stdout);
17015     factorial();
17016     tt{
17017         chal_bsdk();
17018     }
17019     return 0;
17020 }

```

```

17021
17022 //RJBias
17023 // Code by Reyaan Jagnani
17024 #include<bits/stdc++.h>
17025 #define ll long long int
17026 #define ld long double
17027 #define ff first
17028 #define ss second
17029 #define all(x) (x).begin(), (x).end()
17030 #define scanit(a,n) for(ll indexaa=0; indexaa<n; indexaa++) cin>>a[indexaa];
17031 #define printit(a,n) for(ll indexaa=0; indexaa<n; indexaa++) cout<<a[indexaa]<<" ";
17032 cout<<endl;
17033 #define pb push_back
17034 #define precision(a) cout<<fixed<<setprecision(a)
17035 #define testcase ll t; cin>>t; while(t--)
17036 #define endl "\n"
17037 #define iendl "\n", cout<<flush // FOR INTERACTIVE PROBLEMS
17038 #define quick ios_base::sync_with_stdio(false); cin.tie(NULL); cout.tie(NULL)
17039 #define timetaken cerr<<fixed<<setprecision(10); cerr << "time taken : " <<
17040 (float)clock() / CLOCKS_PER_SEC << " secs" << endl
17041 using namespace std;
17042 const ll M = 1000000007;
17043 const ll maxN = 200001;
17044 ll fact[maxN] = {}, smallestPrimeFactor[maxN] = {}, isPrimeSieve[maxN] = {};
17045 mt19937_64 my_rand(chrono::steady_clock::now().time_since_epoch().count());
17046 inline bool comp(ll x,ll y) { return x<y; } // INITIALLY IN DEFAULT INCREASING ORDER
17047 (SMALL TO BIG)
17048 inline ll mod(ll x) {ll a1=(x%M); if(a1<0){a1+=M;} return a1;}
17049 inline ll power(ll x, unsigned ll y, ll p = LLONG_MAX) {ll res=1; x=x%p; if(x==0) {
17050 return 0;} while(y>0){ if(y&1){res=(res*x)%p;} y=y>>1; x=(x*x)%p;} return res;} //
17051 CALCULATING POWER IN LOG(Y) TIME COMPLEXITY
17052 inline ll inversePrimeModular(ll a, ll p) {return power(a,p-2,p);}
17053 inline void calcFact(ll n = maxN-1) { fact[0] = 1; for(ll i=1; i<=n; i++){ fact[i] =
17054 fact[i-1]*i; fact[i] = mod(fact[i]); }}
17055 inline ll ncr(ll n, ll r) { if(n<r) return 0; return mod(inversePrimeModular(mod(fact[n-
17056 r]*fact[r]),M)*fact[n]); }
17057 inline ll ceil(ll a, ll b) { if(b==0) return LLONG_MAX; ll ans = (a+b-1)/b; return ans; }
17058 struct custom_hash { static uint64_t splitmix64(uint64_t x) { x += 0x9e3779b97f4a7c15;
17059 x = (x ^ (x >> 30)) * 0xbf58476d1ce4e5b9; x = (x ^ (x >> 27)) * 0x94d049bb133111eb;
17060 return x ^ (x >> 31); } size_t operator()(uint64_t x) const { static const uint64_t
17061 FIXED_RANDOM = chrono::steady_clock::now().time_since_epoch().count(); return splitmix64
17062 (x + FIXED_RANDOM); }};
17063 void sieve(ll n = maxN-1) { for(ll i=1; i<=n; i++) smallestPrimeFactor[i] = i; for(ll i=
17064 2; (i*i)<=n; i++) { if(smallestPrimeFactor[i]==i) { for(ll j=(i*i); j<=n; j+=i) {
17065 smallestPrimeFactor[j] = min(smallestPrimeFactor[j], i); } } } for(ll i=2; i<=n; i++) {
17066 if(smallestPrimeFactor[i]==i) isPrimeSieve[i] = 1; } }
17067 #ifndef ONLINE_JUDGE
17068 #define dbg(x) cerr << #x << " : "; _print_(x);cerr << endl;
17069 #else
17070 #define dbg(x)
17071 #endif
17072 void _print_(ll t) {cerr << t;}
17073 void _print_(int t) {cerr << t;}
17074 void _print_(string t) {cerr << t;}
17075 void _print_(char t) {cerr << t;}
17076 void _print_(ld t) {cerr << t;}
17077 void _print_(double t) {cerr << t;}
17078 template <class T, class V> void _print_(pair <T, V> p);
17079 template <class T> void _print_(vector <T> v);
17080 template <class T> void _print_(set <T> v);
17081 template <class T, class V> void _print_(map <T, V> v);
17082 template <class T> void _print_(multiset <T> v);
17083 template <class T, class V> void _print_(pair <T, V> p) {cerr << "{"; _print_(p.ff);
17084 cerr << ","; _print_(p.ss); cerr << "}";}
17085 template <class T> void _print_(vector <T> v) {cerr << "["; for (T i : v) {_print_(i);
17086 cerr << " ";} cerr << "];"}
17087 template <class T> void _print_(set <T> v) {cerr << "["; for (T i : v) {_print_(i);
17088 cerr << " ";} cerr << "];"}
17089 template <class T> void _print_(multiset <T> v) {cerr << "["; for (T i : v) {_print_(i

```

```

17073     }; cerr << " ";} cerr << "];}
17074 template <class T, class V> void _print_(map <T, V> v) {cerr << "[ "; for (auto i : v) {
17075     _print_(i); cerr << " ";} cerr << "];}
17076 long long readInt(long long l,long long r,char endd){
17077     long long x=0;
17078     int cnt=0;
17079     int fi=-1;
17080     bool is_neg=false;
17081     while(true){
17082         char g=getchar();
17083         if(g=='-'){
17084             assert(fi==-1);
17085             is_neg=true;
17086             continue;
17087         }
17088         if('0'<=g && g<='9'){
17089             x*=10;
17090             x+=g-'0';
17091             if(cnt==0){
17092                 fi=g-'0';
17093             }
17094             cnt++;
17095             assert(fi!=0 || cnt==1);
17096             assert(fi!=0 || is_neg==false);
17097
17098             assert(!(cnt>19 || ( cnt==19 && fi>1) ));
17099         } else if(g==endd){
17100             if(is_neg){
17101                 x= -x;
17102             }
17103             assert(l<=x && x<=r);
17104             return x;
17105         } else {
17106             assert(false);
17107         }
17108     }
17109 }
17110 string readString(int l,int r,char endd){
17111     string ret="";
17112     int cnt=0;
17113     while(true){
17114         char g=getchar();
17115         assert(g!=-1);
17116         if(g==endd){
17117             break;
17118         }
17119         cnt++;
17120         ret+=g;
17121     }
17122     assert(l<=cnt && cnt<=r);
17123     return ret;
17124 }
17125 long long readIntSp(long long l,long long r){
17126     return readInt(l,r,' ');
17127 }
17128 long long readIntLn(long long l,long long r){
17129     return readInt(l,r,'\n');
17130 }
17131 string readStringLn(int l,int r){
17132     return readString(l,r,'\n');
17133 }
17134 string readStringSp(int l,int r){
17135     return readString(l,r,' ');
17136 }
17137 void case2(vector<ll> &vect, vector<ll> &final, vector<ll> &m, ll n, ll i, ll large)
17138 {
17139     for(ll j=0; j<i; j++)
17140         final.pb(vect[j]);
17141     for(ll j=large-1; j>vect[i]; j--)

```

```

17140     {
17141         if(!m[j])
17142             large = j;
17143     }
17144     final.pb(large);
17145     m[large] = 1;
17146     for(ll i=0; i<m.size(); i++)
17147     {
17148         if(!m[i])
17149             final.pb(i);
17150     }
17151 }
17152 ll solve(vector<ll> &vect, vector<ll> &final, ll p)
17153 {
17154     ll ans = 0;
17155     reverse(all(vect));
17156     reverse(all(final));
17157     while(final.size() > vect.size())
17158         vect.pb(0);
17159     while(final.size() < vect.size())
17160         final.pb(0);
17161     for(ll i=0; i<vect.size(); i++)
17162     {
17163         ll temp1 = mod(final[i]*power(p,i,M));
17164         ll temp2 = mod(vect[i]*power(p,i,M));
17165         ans += mod(temp1 - temp2);
17166         ans = mod(ans);
17167     }
17168     return ans;
17169 }
17170 int main()
17171 {
17172     quick;
17173 #ifndef ONLINE_JUDGE
17174     freopen("edge.in", "r", stdin);
17175     freopen("edge.out", "w", stdout);
17176     // freopen("error.txt", "w", stderr);
17177 #endif
17178     ll sum = 0, k = 2;
17179     ll T = readIntLn(1,1e4);
17180     while(T--)
17181     {
17182         // dbg(k);
17183         k+=2;
17184         ll n = readIntSp(1,1e6);
17185         ll p = readIntLn(1,1e6);
17186         sum += p;
17187         assert(n<=p);
17188         vector<ll> vect(n);
17189         for(ll i=0; i<n-1; i++)
17190             vect[i] = readIntSp(0,p-1);
17191         vect[n-1] = readIntLn(0,p-1);
17192         assert(vect[0]!=0);
17193         vector<ll> m(p), final;
17194         if(n<p)
17195         {
17196             final.pb(1);
17197             final.pb(0);
17198             for(ll i=2; i<=p-1; i++)
17199                 final.pb(i);
17200             cout<<solve(vect, final, p)<<endl;
17201             continue;
17202         }
17203         ll large = p-1, i = 0;
17204         bool check = 1;
17205         while(i<n)
17206         {
17207             if(m[vect[i]])
17208             {

```

```

17209         while(large>=0 && m[large])
17210             large--;
17211         if(large < vect[i])
17212         {
17213             i--;
17214             while(i>=0 && large < vect[i])
17215             {
17216                 m[vect[i]] = 0;
17217                 large = max(large, vect[i]);
17218                 i--;
17219             }
17220             if(i<0)
17221             {
17222                 final.pb(1);
17223                 final.pb(0);
17224                 final.pb(0);
17225                 for(ll j=2; j<p; j++)
17226                     final.pb(j);
17227             }
17228             else
17229             {
17230                 m[vect[i]] = 0;
17231                 case2(vect, final, m, n, i, large);
17232             }
17233         }
17234         else
17235             case2(vect, final, m, n, i, large);
17236         check = 0;
17237         break;
17238     }
17239     else
17240         m[vect[i]] = 1;
17241     i++;
17242 }
17243 if(check)
17244     cout<<"0"<<endl;
17245 else
17246     cout<<solve(vect, final, p)<<endl;
17247 }
17248 assert(sum<=1e6);
17249 assert(getchar()==-1); // Ensures that there are no extra characters at the end.
17250 cerr<<"SUCCESS\n"; // You should see this on the
http://campus.codechef.com/files/stderr/SUBMISSION\_ID page, at the bottom.
17251 timetaken;
17252 return 0;
17253 }
17254 /*
17255
17256
17257 1. Binary Search / Binary Search on Answer
17258 2. Bit
17259 3. Parity (Odd / Even)
17260 4. DP / Greedy
17261 5. Graph / Bi-Partite
17262 */
17263
17264 //PREFIXFLIP
17265 #include <iostream>
17266 #include <string>
17267 #include <set>
17268 #include <map>
17269 #include <stack>
17270 #include <queue>
17271 #include <vector>
17272 #include <utility>
17273 #include <iomanip>
17274 #include <sstream>
17275 #include <bitset>
17276 #include <cstdlib>

```

```

17277 #include <iterator>
17278 #include <algorithm>
17279 #include <cstdio>
17280 #include <cctype>
17281 #include <cmath>
17282 #include <math.h>
17283 #include <ctime>
17284 #include <cstring>
17285 #include <unordered_set>
17286 #include <unordered_map>
17287 #include <cassert>
17288 #define int long long int
17289 #define pb push_back
17290 #define mp make_pair
17291 #define mod 1000000007
17292 #define vl vector<ll>
17293 #define all(c) (c).begin(), (c).end()
17294 using namespace std;
17295
17296 const int N=500023;
17297 bool vis[N];
17298 vector<int> adj[N];
17299 long long readInt(long long l, long long r, char endd){
17300     long long x=0;
17301     int cnt=0;
17302     int fi=-1;
17303     bool is_neg=false;
17304     while(true){
17305         char g=getchar();
17306         if(g=='-'){
17307             assert(fi==-1);
17308             is_neg=true;
17309             continue;
17310         }
17311         if('0'<=g && g<='9'){
17312             x*=10;
17313             x+=g-'0';
17314             if(cnt==0){
17315                 fi=g-'0';
17316             }
17317             cnt++;
17318             assert(fi!=0 || cnt==1);
17319             assert(fi!=0 || is_neg==false);
17320
17321             assert(!(cnt>19 || (cnt==19 && fi>1)));
17322         } else if(g==endd){
17323             if(is_neg){
17324                 x=-x;
17325             }
17326
17327             if(!(l <= x && x <= r))
17328             {
17329                 cerr << l << ' ' << r << ' ' << x << '\n';
17330                 assert(1 == 0);
17331             }
17332
17333             return x;
17334         } else {
17335             assert(false);
17336         }
17337     }
17338 }
17339 string readString(int l, int r, char endd){
17340     string ret="";
17341     int cnt=0;
17342     while(true){
17343         char g=getchar();
17344         assert(g!=-1);
17345         if(g==endd){

```



```

17346         break;
17347     }
17348     cnt++;
17349     ret+=g;
17350 }
17351 assert(l<=cnt && cnt<=r);
17352 return ret;
17353 }
17354 long long readIntSp(long long l,long long r){
17355     return readInt(l,r,' ');
17356 }
17357 long long readIntLn(long long l,long long r){
17358     return readInt(l,r,'\n');
17359 }
17360 string readStringLn(int l,int r){
17361     return readString(l,r,'\n');
17362 }
17363 string readStringSp(int l,int r){
17364     return readString(l,r,' ');
17365 }
17366
17367 void solve()
17368 {
17369     int n = readIntSp(1, 300000);
17370     int k = readIntLn(1, n);
17371     string s = readStringLn(1, n);
17372     int ans = n;
17373     int cnt = 0;
17374     for(int i = 0; i<k-1; i++){
17375         if(s[i] != s[i+1]){
17376             cnt++;
17377         }
17378     }
17379     if(s[k-1] == '0'){
17380         ans = min(ans, cnt + 1);
17381     }
17382     else{
17383         ans = min(ans, cnt);
17384     }
17385     for(int i = k-1; i<n-1; i++){
17386         if(s[i] != s[i+1]){
17387             cnt++;
17388         }
17389         if(s[i-k+1] != s[i-k+2]){
17390             cnt--;
17391         }
17392         if(s[i+1] == '0'){
17393             ans = min(ans, cnt + 1);
17394         }
17395         else{
17396             ans = min(ans, cnt);
17397         }
17398     }
17399     if(s[n-1] == '0'){
17400         ans = min(ans, cnt + 1);
17401     }
17402     else{
17403         ans = min(ans, cnt);
17404     }
17405     cout << ans;
17406 }
17407 int32_t main()
17408 {
17409     #ifndef ONLINE_JUDGE
17410     freopen("input.txt", "r", stdin);
17411     freopen("output.txt", "w", stdout);
17412     #endif
17413     ios_base::sync_with_stdio(false);
17414     cin.tie(NULL),cout.tie(NULL);

```

```

17415     int T=readInt(1,2000,'\n');
17416     while(T--){
17417         solve();
17418         cout<<'\n';
17419     }
17420     assert(getchar()=='\n');
17421     cerr << "Time : " << 1000 * ((double)clock()) / (double)CLOCKS_PER_SEC << "ms\n";
17422 }
17423
17424 //UNQMODE
17425 #include<bits/stdc++.h>
17426 using namespace std;
17427
17428 #define mod 1000000007
17429 typedef set<string> ss;
17430 typedef vector<int> vs;
17431 typedef map<int, char> msi;
17432 typedef pair<int, int> pa;
17433 typedef long long int ll;
17434
17435 ll n, i, last_pos[100005], lp, a[100005], j, ans, k=100;
17436 map<ll, ll> pos;
17437 pair<ll, ll> v[100005];
17438 int main()
17439 {
17440     ios_base::sync_with_stdio(false);
17441     cin.tie(0);
17442
17443     int t;
17444     cin >> t;
17445     while (t--)
17446     {
17447         cin >> n;
17448         for (i = 0; i < n; i++)
17449         {
17450             cin >> a[i];
17451             pos[a[i]] = -1;
17452         }
17453         for (i = 2; i <= k + 1; i++)
17454         {
17455             v[i] = { -1, -1};
17456         }
17457         for (i = 0; i < n; i++)
17458         {
17459             last_pos[i] = pos[a[i]];
17460             pos[a[i]] = i;
17461         }
17462         ans = n;
17463         for (i = 0; i < n; i++)
17464         {
17465             lp = last_pos[i];
17466             for (j = 2; j <= k; j++)
17467             {
17468                 if (lp == -1)
17469                     break;
17470                 if (lp > v[j].second)
17471                 {
17472                     v[j].first = v[j].second;
17473                     v[j].second = lp;
17474                 }
17475                 else if (lp > v[j].first)
17476                 {
17477                     v[j].first = lp;
17478                 }
17479                 lp = last_pos[lp];
17480             }
17481             for (j = 2; j <= k; j++)
17482             {
17483                 ans += v[j].second - max(v[j].first, v[j + 1].second);

```

```

17484     }
17485     }
17486     cout << ans << "\n";
17487 }
17488
17489     return 0;
17490 }
17491
17492 //EXPECTEDSUM
17493 //Utkarsh.25dec
17494 #include <iostream>
17495 #include <cstdio>
17496 #include <cstdlib>
17497 #include <algorithm>
17498 #include <cmath>
17499 #include <vector>
17500 #include <set>
17501 #include <map>
17502 #include <unordered_set>
17503 #include <unordered_map>
17504 #include <queue>
17505 #include <ctime>
17506 #include <cassert>
17507 #include <complex>
17508 #include <string>
17509 #include <cstring>
17510 #include <chrono>
17511 #include <random>
17512 #include <bitset>
17513 #include <array>
17514 #define ll long long int
17515 #define pb push_back
17516 #define mp make_pair
17517 #define mod 998244353
17518 #define vl vector<ll>
17519 #define all(c) (c).begin(), (c).end()
17520 using namespace std;
17521 ll power(ll a, ll b) {ll res=1; a%=mod; assert(b>=0); for(; b; b>>=1) {if(b&1) res=res*a%mod; a
=a*a%mod;} return res;}
17522 ll modInverse(ll a) {return power(a, mod-2);}
17523 const int N=500023;
17524 bool vis[N];
17525 vector<int> adj[N];
17526 long long readInt(long long l, long long r, char endd) {
17527     long long x=0;
17528     int cnt=0;
17529     int fi=-1;
17530     bool is_neg=false;
17531     while(true) {
17532         char g=getchar();
17533         if(g=='-') {
17534             assert(fi==-1);
17535             is_neg=true;
17536             continue;
17537         }
17538         if('0'<=g && g<='9') {
17539             x*=10;
17540             x+=g-'0';
17541             if(cnt==0) {
17542                 fi=g-'0';
17543             }
17544             cnt++;
17545             assert(fi!=0 || cnt==1);
17546             assert(fi!=0 || is_neg==false);
17547
17548             assert(!(cnt>19 || (cnt==19 && fi>1)));
17549         } else if(g==endd) {
17550             if(is_neg) {
17551                 x= -x;

```

```

17552     }
17553
17554     if(!(l <= x && x <= r))
17555     {
17556         cerr << l << ' ' << r << ' ' << x << '\n';
17557         assert(1 == 0);
17558     }
17559
17560     return x;
17561 } else {
17562     assert(false);
17563 }
17564 }
17565 }
17566 string readString(int l,int r,char endd){
17567     string ret="";
17568     int cnt=0;
17569     while(true){
17570         char g=getchar();
17571         assert(g!=-1);
17572         if(g==endd){
17573             break;
17574         }
17575         cnt++;
17576         ret+=g;
17577     }
17578     assert(l<=cnt && cnt<=r);
17579     return ret;
17580 }
17581 long long readIntSp(long long l,long long r){
17582     return readInt(l,r,' ');
17583 }
17584 long long readIntLn(long long l,long long r){
17585     return readInt(l,r,'\n');
17586 }
17587 string readStringLn(int l,int r){
17588     return readString(l,r,'\n');
17589 }
17590 string readStringSp(int l,int r){
17591     return readString(l,r,' ');
17592 }
17593 void solve()
17594 {
17595     ll A, B;
17596     A=readInt(0,1000000000,' ');
17597     B=readInt(0,1000000000,'\n');
17598     assert(A+B>=1 && A+B<=1000000000);
17599     ll totalSum=A;
17600     if(A%2 == B%2)
17601     {
17602         ll ans=totalSum*modInverse(2);
17603         ans%=mod;
17604         cout<<ans<<'\n';
17605         return;
17606     }
17607     if(A==0)
17608     {
17609         cout<<0<<'\n';
17610         return;
17611     }
17612     if(B==0)
17613     {
17614         cout<<(A+1)/2<<'\n';
17615         return;
17616     }
17617     ll l=1+(A-1)*modInverse(2);
17618     l%=mod;
17619     l*=A;
17620     l%=mod;

```

```

17621     l+=((B*A)%mod*modInverse(2));
17622     l%=mod;
17623     l*=modInverse(A+B);
17624     l%=mod;
17625     cout<<l<<'\n';
17626 }
17627 int main()
17628 {
17629     #ifndef ONLINE_JUDGE
17630     freopen("input.txt", "r", stdin);
17631     freopen("output.txt", "w", stdout);
17632     #endif
17633     ios_base::sync_with_stdio(false);
17634     cin.tie(NULL),cout.tie(NULL);
17635     int T=readInt(1,1000,'\n');
17636     while(T--)
17637         solve();
17638     assert(getchar()==-1);
17639     cerr << "Time : " << 1000 * ((double)clock()) / (double)CLOCKS_PER_SEC << "ms\n";
17640 }
17641
17642 //MAXLCS
17643 #include <bits/stdc++.h>
17644 #define maxn 5007
17645 using namespace std;
17646
17647 int dp[maxn][maxn];
17648
17649 int main() {
17650     //freopen("inp9.in", "r", stdin);
17651     //freopen("inp9.out", "w", stdout);
17652     int t;
17653     cin >> t;
17654     int sm = 0;
17655     while(t--) {
17656         int n;
17657         cin >> n;
17658         sm += n;
17659         assert(sm <= 5000);
17660         string s;
17661         cin >> s;
17662         string t = s;
17663         reverse(t.begin(), t.end());
17664         for(int i = 0; i < n; i++)
17665             for(int j = 0; j < n; j++)
17666                 dp[i][j] = 0;
17667         for(int i = 0; i < n; i++) {
17668             for(int j = 0; j < n; j++) {
17669                 int now = (s[i] == t[j]), mx = 0, nc = 0;
17670                 if(i > 0) mx = max(dp[i - 1][j], mx);
17671                 if(j > 0) mx = max(dp[i][j - 1], mx);
17672                 if(i > 0 && j > 0) nc = dp[i - 1][j - 1];
17673                 dp[i][j] = max(nc + now, mx);
17674             }
17675         }
17676         cout << dp[n - 1][n - 1]/2 << "\n";
17677     }
17678 }
17679
17680 //SMALLESDIFF
17681 //Utkarsh.25dec
17682 #include <iostream>
17683 #include <cstdio>
17684 #include <cstdlib>
17685 #include <algorithm>
17686 #include <cmath>
17687 #include <vector>
17688 #include <set>
17689 #include <map>

```

```

17690 #include <unordered_set>
17691 #include <unordered_map>
17692 #include <queue>
17693 #include <ctime>
17694 #include <cassert>
17695 #include <complex>
17696 #include <string>
17697 #include <cstring>
17698 #include <chrono>
17699 #include <random>
17700 #include <bitset>
17701 #include <array>
17702 #define ll long long int
17703 #define pb push_back
17704 #define mp make_pair
17705 #define mod 1000000007
17706 #define vl vector<ll>
17707 #define all(c) (c).begin(), (c).end()
17708 using namespace std;
17709 ll power(ll a, ll b) {ll res=1; a%=mod; assert(b>=0); for(; b>=1; {if(b&1) res=res*a%mod; a
=a*a%mod; } return res; }
17710 ll modInverse(ll a) {return power(a, mod-2); }
17711 const int N=1000023;
17712 bool vis[N];
17713 vector<int> adj[N];
17714 long long readInt(long long l, long long r, char endd) {
17715     long long x=0;
17716     int cnt=0;
17717     int fi=-1;
17718     bool is_neg=false;
17719     while(true) {
17720         char g=getchar();
17721         if(g=='-') {
17722             assert(fi==-1);
17723             is_neg=true;
17724             continue;
17725         }
17726         if('0'<=g && g<='9') {
17727             x*=10;
17728             x+=g-'0';
17729             if(cnt==0) {
17730                 fi=g-'0';
17731             }
17732             cnt++;
17733             assert(fi!=0 || cnt==1);
17734             assert(fi!=0 || is_neg==false);
17735
17736             assert(!(cnt>19 || ( cnt==19 && fi>1) ));
17737         } else if(g==endd) {
17738             if(is_neg) {
17739                 x=-x;
17740             }
17741
17742             if(!(l <= x && x <= r))
17743             {
17744                 cerr << l << ' ' << r << ' ' << x << '\n';
17745                 assert(1 == 0);
17746             }
17747
17748             return x;
17749         } else {
17750             assert(false);
17751         }
17752     }
17753 }
17754 string readString(int l, int r, char endd) {
17755     string ret="";
17756     int cnt=0;
17757     while(true) {

```

```

17758         char g=getchar();
17759         assert(g!=-1);
17760         if(g==endd){
17761             break;
17762         }
17763         cnt++;
17764         ret+=g;
17765     }
17766     assert(l<=cnt && cnt<=r);
17767     return ret;
17768 }
17769 long long readIntSp(long long l,long long r){
17770     return readInt(l,r,' ');
17771 }
17772 long long readIntLn(long long l,long long r){
17773     return readInt(l,r,'\n');
17774 }
17775 string readStringLn(int l,int r){
17776     return readString(l,r,'\n');
17777 }
17778 string readStringSp(int l,int r){
17779     return readString(l,r,' ');
17780 }
17781 int sumNM=0;
17782 int B[N];
17783 set<int> s;
17784 int n,m;
17785 int A[1005][1005];
17786 ll ans;
17787 void fillGrid()
17788 {
17789     int blockers=n-1;
17790     int mini=(blockers+1)/2;
17791     int maxi=blockers/2;
17792     int pathLen=n+m-1;
17793     int optL=0, optR=0;
17794     for(int i=mini+1;i<=n*m;i++)
17795     {
17796         int j=i+pathLen-1;
17797         int rem=n*m-j;
17798         if(rem<maxi)
17799             break;
17800         if(ans>B[j]-B[i])
17801         {
17802             ans=B[j]-B[i];
17803             optL=i;
17804             optR=j;
17805         }
17806     }
17807     int curr=optL;
17808     for(int i=1;i<=n;i++)
17809     {
17810         A[i][1]=B[curr++];
17811         s.erase(A[i][1]);
17812     }
17813     for(int j=2;j<=m;j++)
17814     {
17815         A[n][j]=B[curr++];
17816         s.erase(A[n][j]);
17817     }
17818     int lar=n*m;
17819     int sm=1;
17820     for(int i=1;i<=n;i++)
17821     {
17822         if(i%2==1)
17823             A[i][2]=B[sm++];
17824         else
17825             A[i][2]=B[lar--];
17826         s.erase(A[i][2]);

```

```

17827     }
17828     for(int i=1;i<n;i++)
17829     {
17830         for(int j=3;j<=m;j++)
17831         {
17832             A[i][j]=(*s.begin());
17833             s.erase(s.begin());
17834         }
17835     }
17836 }
17837 void solve()
17838 {
17839     s.clear();
17840     ans=1e18;
17841
17842     n=readInt(2,1000,' ');
17843     m=readInt(2,1000,'\n');
17844     sumNM+=(n*m);
17845     assert(sumNM<=1000000);
17846     for(int i=1;i<=n*m;i++)
17847     {
17848         if(i==n*m)
17849             B[i]=readInt(1,1000000000,'\n');
17850         else
17851             B[i]=readInt(1,1000000000,' ');
17852         s.insert(B[i]);
17853     }
17854     assert(s.size()==n*m);
17855     sort(B, B+n*m+1);
17856     if(n<=m)
17857     {
17858         fillGrid();
17859     }
17860     else
17861     {
17862         swap(n,m);
17863         fillGrid();
17864         swap(n,m);
17865         int C[n+1][m+1];
17866         for(int i=1;i<=n;i++)
17867             for(int j=1;j<=m;j++)
17868                 C[i][j]=A[j][i];
17869         for(int i=1;i<=n;i++)
17870             for(int j=1;j<=m;j++)
17871                 A[i][j]=C[i][j];
17872     }
17873     // cout<<ans<<'\n';
17874     for(int i=1;i<=n;i++)
17875     {
17876         for(int j=1;j<=m;j++)
17877             cout<<A[i][j]<<' ';
17878         cout<<'\n';
17879     }
17880 }
17881 int main()
17882 {
17883     #ifndef ONLINE_JUDGE
17884     freopen("input.txt", "r", stdin);
17885     freopen("output.txt", "w", stdout);
17886     #endif
17887     ios_base::sync_with_stdio(false);
17888     cin.tie(NULL),cout.tie(NULL);
17889     int T=readInt(1,10000,'\n');
17890     while(T--)
17891         solve();
17892     assert(getchar()==-1);
17893     cerr << "Time : " << 1000 * ((double)clock()) / (double)CLOCKS_PER_SEC << "ms\n";
17894 }
17895

```



```

17896 //FREQARRET
17897 #include <iostream>
17898 #include <map>
17899
17900 using namespace std;
17901
17902 map<int,int> F;
17903 pair<int, int> M[100010];
17904 int q, n, B[100010], poss, prev1;
17905
17906 int main()
17907 {
17908     cin>>q;
17909
17910     while(q-->0)
17911     {
17912         F.clear();
17913         poss = 1;
17914         prev1 = 0;
17915         cin>>n;
17916         int cnt = 0;
17917         for(int i=1;i<=n;i++)
17918         {
17919             cin>>B[i];
17920             F[B[i]]++;
17921             M[B[i]].first = 0;
17922             M[B[i]].second = 0;
17923         }
17924
17925         for(auto it = F.begin(); it != F.end(); it++)
17926         {
17927             if( ((it->second)%(it->first))!=0 )
17928             {
17929                 poss = 0;
17930                 break;
17931             }
17932         }
17933         if(poss==0)
17934         {
17935             cout<<-1<<"\n";
17936             continue;
17937         }
17938
17939         for(int i=1;i<=n;i++)
17940         {
17941             if( (M[B[i]].first == 0) || (M[B[i]].second == B[i]) )
17942             {
17943                 prev1++;
17944                 M[B[i]].first = prev1;
17945                 M[B[i]].second = 1;
17946             }
17947             else
17948                 M[B[i]].second++;
17949             cout<<M[B[i]].first;
17950             if(i<n)
17951                 cout<<" ";
17952             else
17953                 cout<<"\n";
17954         }
17955     }
17956 }
17957
17958 //TWO COUNTERS
17959 #include <bits/stdc++.h>
17960 #include <ext/pb_ds/assoc_container.hpp>
17961 #include <ext/pb_ds/tree_policy.hpp>
17962 #include <ext/rope>
17963 using namespace std;
17964 using namespace __gnu_pbds;

```

```

17965 using namespace __gnu_cxx;
17966
17967 #define int long long
17968 #define ll long long
17969 #define ii pair<ll,ll>
17970 #define iii pair<ii,ll>
17971 #define fi first
17972 #define se second
17973 #define endl '\n'
17974 #define debug(x) cout << #x << ": " << x << endl
17975
17976 #define pub push_back
17977 #define pob pop_back
17978 #define puf push_front
17979 #define pof pop_front
17980 #define lb lower_bound
17981 #define ub upper_bound
17982
17983 #define rep(x,start,end) for(auto
x=(start)-((start)>(end));x!=(end)-((start)>(end));((start)<(end)?x++:x--))
17984 #define all(x) (x).begin(),(x).end()
17985 #define sz(x) (int)(x).size()
17986
17987 #define indexed_set
tree<ll,null_type,less<ll>,rb_tree_tag,tree_order_statistics_node_update>
17988 //change less to less_equal for non distinct pbds, but erase will bug
17989
17990 mt19937 rng(chrono::system_clock::now().time_since_epoch().count());
17991
17992 int n,m;
17993 int arr[100005];
17994 int brr[100005];
17995 int typ[100005];
17996 int dp[2][5];
17997
17998 signed main(){
17999     ios::sync_with_stdio(0);
18000     cin.tie(0);
18001     cout.tie(0);
18002     cin.exceptions(ios::badbit | ios::failbit);
18003
18004     int TC;
18005     cin>>TC;
18006     while (TC--){
18007         cin>>n>>m;
18008
18009         rep(x,1,n+1) typ[x]=-1;
18010         rep(x,0,m) cin>>arr[x];
18011         rep(x,0,m) cin>>brr[x];
18012         rep(x,0,m) typ[arr[x]]=brr[x];
18013
18014         int a=0,b=1;
18015
18016         memset(dp,128,sizeof(dp));
18017         dp[a][2]=0;
18018
18019         rep(x,1,n+1){
18020             memset(dp[b],128,sizeof(dp[b]));
18021             rep(x,0,5){
18022                 if (x!=0) dp[b][x-1]=max(dp[b][x-1],dp[a][x]);
18023                 if (x!=4) dp[b][x+1]=max(dp[b][x+1],dp[a][x]);
18024             }
18025
18026             if (typ[x]==1){
18027                 dp[b][2]=max({dp[b][0],dp[b][1],dp[b][2]});
18028                 dp[b][0]=-1e9;
18029                 dp[b][1]=-1e9;
18030                 dp[b][3]++;
18031                 dp[b][4]++;

```

```

18032     }
18033     if (typ[x]==2) {
18034         dp[b][2]=max({dp[b][2],dp[b][3],dp[b][4]});
18035         dp[b][0]++;
18036         dp[b][1]++;
18037         dp[b][3]=-1e9;
18038         dp[b][4]=-1e9;
18039     }
18040
18041     swap(a,b);
18042 }
18043
18044 int ans=0;
18045 rep(x,0,5) ans=max(ans,dp[a][x]);
18046 cout<<ans<<endl;
18047 }
18048 }
18049
18050 //OPERATION
18051 #include <map>
18052 #include <set>
18053 #include <cmath>
18054 #include <ctime>
18055 #include <queue>
18056 #include <stack>
18057 #include <cstdio>
18058 #include <cstdlib>
18059 #include <vector>
18060 #include <cstring>
18061 #include <algorithm>
18062 using namespace std;
18063 typedef double db;
18064 typedef long long ll;
18065 typedef unsigned long long ull;
18066 const int N=1000010;
18067 const int LOGN=28;
18068 const ll TMD=0;
18069 const ll INF=2147483647;
18070 int n,q;
18071 int a[N],XOR[N],MX[N],ans[N];
18072 vector<pair<int,int> >query[N];
18073
18074 struct nod
18075 {
18076     nod *ch[2];
18077 };
18078
18079 struct Trie
18080 {
18081     nod *root;
18082
18083     Trie()
18084     {
18085         root=NULL;
18086     }
18087
18088     void newnod(nod **p)
18089     {
18090         *p=new nod;
18091         (*p)->ch[0]=(*p)->ch[1]=NULL;
18092     }
18093
18094     void insert(int x)
18095     {
18096         _insert(&root,x,29);
18097     }
18098
18099     void _insert(nod **p,int x,int b)
18100     {

```

```

18101         if(*p==NULL) newnod(p);
18102         if(b==-1) return ;
18103         _insert(&(*p)->ch[(x&(1<<b))!=0],x,b-1);
18104     }
18105
18106     int getmx(int x)
18107     {
18108         return _getmx(root,x,29,0);
18109     }
18110
18111     int _getmx(nod *p,int x,int b,int cur)
18112     {
18113         if(b==-1) return cur;
18114         int t=(x&(1<<b))!=0;
18115         if(p->ch[t^1]) return _getmx(p->ch[t^1],x,b-1,cur+(1<<b));
18116         else return _getmx(p->ch[t],x,b-1,cur);
18117     }
18118 };
18119
18120 int main()
18121 {
18122     scanf("%d%d",&n,&q);
18123     for(int i=1;i<=n;i++) scanf("%d",&a[i]);
18124     for(int i=n;i;i--) XOR[i]=XOR[i+1]^a[i],MX[i]=max(XOR[i],MX[i+1]);
18125     for(int i=1;i<=q;i++)
18126     {
18127         int p,x;
18128         scanf("%d%d",&p,&x);
18129         ans[i]=MX[p+1];
18130         query[p-1].push_back(make_pair(XOR[1]^a[p]^x,i));
18131     }
18132     int t=0;
18133     Trie T;
18134     for(int i=1;i<=n;i++)
18135     {
18136         t^=a[i];
18137         T.insert(t);
18138         for(int j=0;j<query[i].size();j++)
18139         {
18140             int x=query[i][j].first,id=query[i][j].second;
18141             ans[id]=max(ans[id],T.getmx(x));
18142         }
18143     }
18144     for(int i=1;i<=q;i++) printf("%d\n",ans[i]);
18145
18146     return 0;
18147 }
18148
18149 //OPERATION-EDITORS SOLUTION
18150 #include "bits/stdc++.h"
18151 // #pragma GCC optimize("O3,unroll-loops")
18152 // #pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
18153 using namespace std;
18154 using ll = long long int;
18155 mt19937_64 rng(chrono::high_resolution_clock::now().time_since_epoch().count());
18156
18157 struct Trie {
18158     vector<int> v;
18159     vector<array<int, 2>> ch;
18160     int id = 0;
18161     Trie() : v(1, 0), ch(1, {-1, -1}) {}
18162     void create() {
18163         v.push_back(0);
18164         ch.push_back({-1, -1});
18165         ++id;
18166     }
18167     void add(int x) {
18168         int node = 0;
18169         for (int bit = 30; bit >= 0; --bit) {

```

```

18170         int b = (x >> bit) & 1;
18171         ++v[node];
18172         if (ch[node][b] == -1) {
18173             create();
18174             ch[node][b] = id;
18175         }
18176         node = ch[node][b];
18177     }
18178     ++v[node];
18179 }
18180 int query (int x) { // Maximum value of a^x for a in the trie
18181     int node = 0, ret = 0;
18182     for (int bit = 30; bit >= 0; --bit) {
18183         int b = (x >> bit) & 1;
18184         if (ch[node][b^1] == -1) node = ch[node][b];
18185         else {
18186             ret += 1 << bit;
18187             node = ch[node][b^1];
18188         }
18189     }
18190     return ret;
18191 }
18192 };
18193
18194 int main()
18195 {
18196     ios::sync_with_stdio(false); cin.tie(0);
18197
18198     int n, q; cin >> n >> q;
18199     vector<int> a(n);
18200     for (int &x : a) cin >> x;
18201     vector<int> suf(n), sufmax(n+1);
18202     for (int i = n-1; i >= 0; --i) {
18203         suf[i] = a[i];
18204         if (i < n-1) suf[i] ^= suf[i+1];
18205         sufmax[i] = suf[i];
18206         if (i < n-1) sufmax[i] = max(sufmax[i], sufmax[i+1]);
18207     }
18208
18209     vector<vector<array<int, 2>>> queries(n);
18210     vector<int> ans(q);
18211     for (int i = 0; i < q; ++i) {
18212         int pos, val; cin >> pos >> val;
18213         queries[--pos].push_back({val, i});
18214         ans[i] = sufmax[pos+1];
18215     }
18216
18217     Trie T;
18218
18219     for (int i = 1; i < n; ++i) {
18220         T.add(suf[i]);
18221         for (auto [val, id] : queries[i]) {
18222             ans[id] = max(ans[id], T.query(val ^ a[i]));
18223         }
18224     }
18225     for (auto x : ans) cout << x << '\n';
18226 }
18227
18228 //XYTREE
18229
18230 /* Author : Chaitanya Darwai */
18231 #include <bits/stdc++.h>
18232 using namespace std;
18233
18234 #define rep(i,a,b,c)          for(int i=a; i<b; i+=c)
18235 #define rrep(i,a,b)           for(int i=b-1; i>=a; i--)
18236 #define vec                   vector
18237 typedef vec<int>               vi;
18238 typedef pair<int,int>          pii;

```

```

18239 const int P = 1e9+7;
18240 // const int P = 998244353;
18241 const int N = 1e5+1;
18242
18243 void solve(int tcn){
18244     int n;
18245     cin >> n;
18246
18247     vec<vi> g(n);
18248
18249     for(int i = 0; i < n-1; ++i){
18250         int u, v;
18251         cin >> u >> v;
18252         --u, --v;
18253         g[u].push_back(v);
18254         g[v].push_back(u);
18255     }
18256
18257     vi num(n), val(n, 0), par(n);
18258
18259     num[0] = g[0].size();
18260     for(int i = 1; i < n; ++i){
18261         num[i] = g[i].size()-1;
18262     }
18263
18264     auto dfs = [&](int u, int _par, auto &dfs)->void{
18265         par[u] = _par;
18266         for(auto v : g[u]) if(v != _par) dfs(v, u, dfs);
18267     };
18268     dfs(0, -1, dfs);
18269
18270     int q;
18271     cin >> q;
18272
18273     int ans = n-1;
18274     while(q--){
18275         int type;
18276         cin >> type;
18277
18278         if(type == 2){
18279             cout << ans << '\n';
18280         }
18281         else{
18282             int u;
18283             cin >> u;
18284             --u;
18285             if(!val[u] and !num[u]){
18286                 if(u) --num[par[u]];
18287                 val[u] = 1;
18288                 --ans;
18289             }
18290             if(!u and val[u]) ans = n-1;
18291         }
18292     }
18293 }
18294
18295 signed main(){
18296     ios_base::sync_with_stdio(0); cin.tie(NULL); cout.tie(NULL);
18297
18298     int test = 1;
18299     cin >> test;
18300     rep(i,0,test,1){
18301         solve(i+1);
18302     }
18303     return 0;
18304 }
18305
18306 //XYTREE-EDITORIAL
18307 #include "bits/stdc++.h"

```

```

18308 // #pragma GCC optimize("O3,unroll-loops")
18309 // #pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
18310 using namespace std;
18311 using ll = long long int;
18312 mt19937_64 rng(chrono::high_resolution_clock::now().time_since_epoch().count());
18313
18314 int main()
18315 {
18316     ios::sync_with_stdio(false); cin.tie(0);
18317
18318     int t; cin >> t;
18319     while (t--) {
18320         int n; cin >> n;
18321         vector<vector<int>> g(n);
18322         for (int i = 0; i < n-1; ++i) {
18323             int u, v; cin >> u >> v;
18324             g[--u].push_back(--v);
18325             g[v].push_back(u);
18326         }
18327
18328         vector<int> par(n), unchanged(n), a(n);
18329         auto dfs = [&] (const auto &self, int u, int p) -> void {
18330             par[u] = p;
18331             for (int v : g[u]) {
18332                 if (v == p) continue;
18333                 self(self, v, u);
18334                 ++unchanged[u];
18335             }
18336         };
18337         dfs(dfs, 0, 0);
18338
18339         int q; cin >> q;
18340         int ans = n-1;
18341         while (q--) {
18342             int type; cin >> type;
18343             if (type == 1) {
18344                 int u; cin >> u; --u;
18345                 if (a[u] == 0 and unchanged[u] == 0) {
18346                     a[u] = 1;
18347                     --ans;
18348                     if (u == 0) ans = n-1;
18349                     else --unchanged[par[u]];
18350                 }
18351             }
18352             else cout << ans << '\n';
18353         }
18354     }
18355 }
18356
18357 //PRIMEREVERSE
18358 #include <iostream>
18359 #include <string>
18360 #include <set>
18361 #include <map>
18362 #include <stack>
18363 #include <queue>
18364 #include <vector>
18365 #include <utility>
18366 #include <iomanip>
18367 #include <sstream>
18368 #include <bitset>
18369 #include <cstdlib>
18370 #include <iterator>
18371 #include <algorithm>
18372 #include <cstdio>
18373 #include <cctype>
18374 #include <cmath>
18375 #include <math.h>
18376 #include <ctime>

```

```

18377 #include <cstring>
18378 #include <unordered_set>
18379 #include <unordered_map>
18380 #include <cassert>
18381 #define int long long int
18382 #define pb push_back
18383 #define mp make_pair
18384 #define mod 1000000007
18385 #define vl vector<ll>
18386 #define all(c) (c).begin(), (c).end()
18387 using namespace std;
18388
18389 const int N=500023;
18390 bool vis[N];
18391 vector<int> adj[N];
18392 long long readInt(long long l, long long r, char endd){
18393     long long x=0;
18394     int cnt=0;
18395     int fi=-1;
18396     bool is_neg=false;
18397     while(true){
18398         char g=getchar();
18399         if(g=='-'){
18400             assert(fi==-1);
18401             is_neg=true;
18402             continue;
18403         }
18404         if('0'<=g && g<='9'){
18405             x*=10;
18406             x+=g-'0';
18407             if(cnt==0){
18408                 fi=g-'0';
18409             }
18410             cnt++;
18411             assert(fi!=0 || cnt==1);
18412             assert(fi!=0 || is_neg==false);
18413
18414             assert(!(cnt>19 || ( cnt==19 && fi>1) ));
18415         } else if(g==endd){
18416             if(is_neg){
18417                 x= -x;
18418             }
18419
18420             if(!(l <= x && x <= r))
18421             {
18422                 cerr << l << ' ' << r << ' ' << x << '\n';
18423                 assert(1 == 0);
18424             }
18425
18426             return x;
18427         } else {
18428             assert(false);
18429         }
18430     }
18431 }
18432 string readString(int l, int r, char endd){
18433     string ret="";
18434     int cnt=0;
18435     while(true){
18436         char g=getchar();
18437         assert(g!=-1);
18438         if(g==endd){
18439             break;
18440         }
18441         cnt++;
18442         ret+=g;
18443     }
18444     assert(l<=cnt && cnt<=r);
18445     return ret;

```



```

18446 }
18447 long long readIntSp(long long l,long long r){
18448     return readInt(l,r,' ');
18449 }
18450 long long readIntLn(long long l,long long r){
18451     return readInt(l,r,'\n');
18452 }
18453 string readStringLn(int l,int r){
18454     return readString(l,r,'\n');
18455 }
18456 string readStringSp(int l,int r){
18457     return readString(l,r,' ');
18458 }
18459
18460 void solve()
18461 {
18462     int n = readIntLn(1, 100000);
18463     string s = readStringLn(1, n);
18464     string t = readStringLn(1, n);
18465
18466     assert(s.size() == n);
18467     assert(t.size() == n);
18468
18469     int s1 = 0, s2 = 0, t1 = 0, t2 = 0;
18470
18471     for(int i = 0; i < n; i++){
18472         if(s[i] == '1')
18473             s1++;
18474         else
18475             s2++;
18476
18477         if(t[i] == '1')
18478             t1++;
18479         else
18480             t2++;
18481     }
18482
18483     if(s1==t1 && s2==t2){
18484         cout << "YES";
18485     }
18486     else{
18487         cout << "NO";
18488     }
18489 }
18490 int32_t main()
18491 {
18492     #ifndef ONLINE_JUDGE
18493     freopen("input.txt", "r", stdin);
18494     freopen("output.txt", "w", stdout);
18495     #endif
18496     ios_base::sync_with_stdio(false);
18497     cin.tie(NULL),cout.tie(NULL);
18498     int T=readInt(1,100,'\n');
18499     while(T--){
18500         solve();
18501         cout<<'\n';
18502     }
18503     assert(getchar()==-1);
18504     cerr << "Time : " << 1000 * ((double)clock()) / (double)CLOCKS_PER_SEC << "ms\n";
18505 }
18506
18507 //SQUARE_LOOP
18508 #include <map>
18509 #include <set>
18510 #include <cmath>
18511 #include <ctime>
18512 #include <queue>
18513 #include <stack>
18514 #include <cstdio>

```

```

18515 #include <cstdlib>
18516 #include <vector>
18517 #include <cstring>
18518 #include <algorithm>
18519 using namespace std;
18520 typedef double db;
18521 typedef long long ll;
18522 typedef unsigned long long ull;
18523 const int N=510;
18524 const int LOGN=11;
18525 const ll TMD=0;
18526 const ll INF=2147483647;
18527 int T,n,q,ans_origin;
18528 int d[N<<1];
18529 int f[N<<1][LOGN];
18530 char c[N<<1][N<<1];
18531 vector<int> G[N<<1];
18532 map<pair<int,int>,int> tag;
18533
18534 int lca(int x,int y)
18535 {
18536     if(d[x]>d[y]) swap(x,y);
18537     for(int i=LOGN-1;i>=0;i--) if(d[f[y][i]]>=d[x]) y=f[y][i];
18538     if(x==y) return x;
18539     for(int i=LOGN-1;i>=0;i--) if(f[x][i]!=f[y][i]) x=f[x][i],y=f[y][i];
18540     return f[x][0];
18541 }
18542
18543 int solve(vector<int> G[],map<pair<int,int>,int> &tag)
18544 {
18545     queue<int> Q;
18546     for(int i=1;i<=n*2;i++) d[i]=f[i][0]=0;
18547     d[1]=1;
18548     Q.push(1);
18549     while(!Q.empty())
18550     {
18551         int x=Q.front();
18552         Q.pop();
18553         for(int i=0;i<G[x].size();i++)
18554         {
18555             int y=G[x][i];
18556             if(!d[y])
18557             {
18558                 f[y][0]=x;
18559                 d[y]=d[x]+1;
18560                 Q.push(y);
18561             }
18562         }
18563     }
18564     for(int i=1;i<LOGN;i++)
18565         for(int j=1;j<=n*2;j++)
18566             f[j][i]=f[f[j][i-1]][i-1];
18567     int mn=INF,U,V;
18568     for(int i=1;i<=n;i++)
18569     {
18570         if(!d[i]) continue;
18571         for(int j=0;j<G[i].size();j++)
18572         {
18573             int u=i,v=G[i][j],l=lca(u,v);
18574             if(l==u||l==v) continue;
18575             if(d[l]==1&& d[u]+d[v]-1<mn)
18576             {
18577                 mn=d[u]+d[v]-1;
18578                 U=u;V=v;
18579             }
18580         }
18581     }
18582     if(mn==INF) return -1;
18583     tag[make_pair(U,V)]=1;

```

```

18584     while(f[U][0]) tag[make_pair(min(U,f[U][0]),max(U,f[U][0]))]=1,U=f[U][0];
18585     while(f[V][0]) tag[make_pair(min(V,f[V][0]),max(V,f[V][0]))]=1,V=f[V][0];
18586     return mn;
18587 }
18588
18589 int main()
18590 {
18591     scanf("%d%d",&n,&q);
18592     for(int i=1;i<=n;i++)
18593     {
18594         scanf("\n");
18595         for(int j=1;j<=n;j++)
18596         {
18597             scanf("%c",&c[i][j]);
18598             if(c[i][j]=='O')
18599             {
18600                 G[i].push_back(j+n);
18601                 G[j+n].push_back(i);
18602             }
18603         }
18604     }
18605     ans_origin=solve(G,tag);
18606     for(int i=1;i<=q;i++)
18607     {
18608         int x,y;
18609         scanf("%d%d",&x,&y);
18610         if(!tag[make_pair(x,y+n)]) printf("%d\n",ans_origin);
18611         else
18612         {
18613             vector<int> G2[N<<1];
18614             map<pair<int,int>,int> tag2;
18615             for(int j=1;j<=n;j++)
18616             {
18617                 for(int k=1;k<=n;k++)
18618                 {
18619                     if(c[j][k]=='O'&&(j!=x||k!=y))
18620                     {
18621                         G2[j].push_back(k+n);
18622                         G2[k+n].push_back(j);
18623                     }
18624                 }
18625             }
18626             printf("%d\n",solve(G2,tag2));
18627         }
18628     }
18629
18630     return 0;
18631 }
18632
18633 //SQUARE_LOOP-EDITOR
18634 #include "bits/stdc++.h"
18635 // #pragma GCC optimize("O3,unroll-loops")
18636 // #pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
18637 using namespace std;
18638 using ll = long long int;
18639 mt19937_64 rng(chrono::high_resolution_clock::now().time_since_epoch().count());
18640
18641 int main()
18642 {
18643     ios::sync_with_stdio(false); cin.tie(0);
18644
18645     int n, q; cin >> n >> q;
18646     vector<string> grid(n);
18647     vector adj(2*n, vector<int>());
18648     for (int i = 0; i < n; ++i) {
18649         cin >> grid[i];
18650         for (int j = 0; j < n; ++j) {
18651             if (grid[i][j] == 'O') {
18652                 adj[i].push_back(n+j);

```

```

18653         adj[n+j].push_back(i);
18654     }
18655 }
18656 }
18657
18658 auto shortest_cycle = [] (auto adj, int src, bool findcyc = false) {
18659     int sz = adj.size();
18660     vector<int> dep(sz, INT_MAX), par(sz, -1);
18661     vector<int> whichch(sz);
18662     queue<int> q; q.push(src); dep[src] = 0;
18663     while (!q.empty()) {
18664         int u = q.front(); q.pop();
18665         for (int v : adj[u]) {
18666             if (dep[v] == INT_MAX) {
18667                 dep[v] = 1 + dep[u];
18668                 par[v] = u;
18669                 q.push(v);
18670                 if (u == src) whichch[v] = v;
18671                 else whichch[v] = whichch[u];
18672             }
18673         }
18674     }
18675     int mincycle = 1e9, x = -1, y = -1;
18676     for (int i = 0; i < sz; ++i) {
18677         for (int u : adj[i]) {
18678             if (i == par[u] or par[i] == u) continue;
18679             if (whichch[i] == whichch[u]) continue;
18680             if (mincycle > dep[i] + dep[u] + 1) {
18681                 mincycle = dep[i] + dep[u] + 1;
18682                 x = i, y = u;
18683             }
18684         }
18685     }
18686     vector<array<int, 2>> edges;
18687     if (mincycle != INT_MAX and findcyc) {
18688         edges.push_back({x, y});
18689         while (x != src) {
18690             edges.push_back({x, par[x]});
18691             x = par[x];
18692         }
18693         while (y != src) {
18694             edges.push_back({y, par[y]});
18695             y = par[y];
18696         }
18697     }
18698     return pair{mincycle, edges};
18699 };
18700
18701 auto [orig_ans, cycle] = shortest_cycle(adj, 0, true);
18702 map<array<int, 2>, int> edge_id;
18703 int id = 1;
18704 for (auto &[x, y] : cycle) {
18705     if (x > y) swap(x, y);
18706     edge_id[{x, y}] = id++;
18707 }
18708 vector<int> ans(id);
18709
18710 for (int i = 1; i < id; ++i) {
18711     auto [u, v] = cycle[i-1];
18712     auto tmp1 = adj[u], tmp2 = adj[v];
18713     vector<int> nw1, nw2;
18714     for (int x : adj[u]) {
18715         if (x != v) nw1.push_back(x);
18716     }
18717     for (int x : adj[v]) {
18718         if (x != u) nw2.push_back(x);
18719     }
18720     adj[u] = nw1; adj[v] = nw2;
18721     ans[i] = shortest_cycle(adj, 0).first;

```

```

18722         adj[u] = tmp1; adj[v] = tmp2;
18723     }
18724
18725     while (q--) {
18726         int x, y; cin >> x >> y;
18727         --x, --y;
18728         int out = orig_ans;
18729         if (edge_id[{x, n+y}]) out = ans[edge_id[{x, n+y}]];
18730         if (out > 1e8) out = -1;
18731         cout << out << '\n';
18732     }
18733 }
18734
18735 //TO START START69
18736 //LONGESTARRAY
18737 #include <bits/stdc++.h>
18738 #include <ext/pb_ds/tree_policy.hpp>
18739 #include <ext/pb_ds/assoc_container.hpp>
18740 using namespace __gnu_pbds;
18741 using namespace std;
18742 #define ll long long
18743 const ll INF_MUL=1e13;
18744 const ll INF_ADD=1e18;
18745 #define pb push_back
18746 #define mp make_pair
18747 #define nline "\n"
18748 #define f first
18749 #define s second
18750 #define pll pair<ll,ll>
18751 #define all(x) x.begin(),x.end()
18752 #define vl vector<ll>
18753 #define vvl vector<vector<ll>>
18754 #define vvvl vector<vector<vector<ll>>>
18755 #ifndef ONLINE_JUDGE
18756 #define debug(x) cerr<<#x<<" "; _print(x); cerr<<nline;
18757 #else
18758 #define debug(x);
18759 #endif
18760 void _print(ll x){cerr<<x;}
18761 void _print(char x){cerr<<x;}
18762 void _print(string x){cerr<<x;}
18763 mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());
18764 template<class T,class V> void _print(pair<T,V> p) {cerr<<"{"; _print(p.first);cerr<<" ";
; _print(p.second);cerr<<"}";}
18765 template<class T>void _print(vector<T> v) {cerr<<" [ "; for (T i:v){_print(i);cerr<<" "
; }cerr<<" ]";}
18766 template<class T>void _print(set<T> v) {cerr<<" [ "; for (T i:v){_print(i); cerr<<" ";}
cerr<<" ]";}
18767 template<class T>void _print(multiset<T> v) {cerr<<" [ "; for (T i:v){_print(i);cerr<<"
"; }cerr<<" ]";}
18768 template<class T,class V>void _print(map<T, V> v) {cerr<<" [ "; for(auto i:v) {_print(i
);cerr<<" ";} cerr<<" ]";}
18769 typedef tree<ll, null_type, less<ll>, rb_tree_tag, tree_order_statistics_node_update>
ordered_set;
18770 typedef tree<ll, null_type, less_equal<ll>, rb_tree_tag,
tree_order_statistics_node_update> ordered_multiset;
18771 typedef tree<pair<ll,ll>, null_type, less<pair<ll,ll>>, rb_tree_tag,
tree_order_statistics_node_update> ordered_pset;
18772 //-----
-----
18773 const ll MOD=998244353;
18774 const ll MAX=500500;
18775 ll freq[35][MAX];
18776 void solve(){
18777     ll n; cin>>n;
18778     vector<ll> till(35,n+1);
18779     vector<ll> a(n+5,0);
18780     ll total=0;

```

```

18781     for(ll i=0;i<30;i++){
18782         freq[i][0]=0;
18783     }
18784     for(ll i=1;i<=n;i++){
18785         ll x; cin>>x;
18786         a[i]=x;
18787         total|=x;
18788         for(ll j=0;j<30;j++){
18789             freq[j][i]=freq[j][i-1]+min(1LL,x&(1<<j));
18790             if(x&(1<<j)){
18791                 till[j]=i;
18792             }
18793         }
18794     }
18795     ll ans=-1,cur=0;
18796     for(ll i=1;i<=n;i++){
18797         ll l=n+1;
18798         for(ll j=0;j<30;j++){
18799             if(!(total&(1<<j))){
18800                 continue;
18801             }
18802             if(cur&(1<<j)){
18803                 ;
18804             }
18805             else{
18806                 l=min(l,till[j]);
18807             }
18808         }
18809         ll check=l>i;
18810         for(ll j=0;j<30;j++){
18811             if(!(total&(1<<j))){
18812                 continue;
18813             }
18814             if(freq[j][i-1] >= freq[j][l-1]){
18815                 check=0;
18816             }
18817         }
18818         if(check){
18819             ans=max(ans,l-i);
18820         }
18821         cur|=a[i];
18822     }
18823     cout<<ans<<endl;
18824     return;
18825 }
18826 int main()

18827 {
18828     ios_base::sync_with_stdio(false);
18829     cin.tie(NULL);
18830     #ifndef ONLINE_JUDGE
18831     freopen("input.txt", "r", stdin);
18832     freopen("output.txt", "w", stdout);
18833     freopen("error.txt", "w", stderr);
18834     #endif
18835     ll test_cases=1;
18836     cin>>test_cases;
18837     while(test_cases--){
18838         solve();
18839     }
18840     cout<<fixed<<setprecision(10);
18841     cerr<<"Time:"<<1000*((double)clock())/(double)CLOCKS_PER_SEC<<"ms\n";
18842 }
18843
18844 //DIVBYTHREE
18845 //Utkarsh.25dec
18846 #include <iostream>
18847 #include <cstdio>

```

```

18848 #include <cstdlib>
18849 #include <algorithm>
18850 #include <cmath>
18851 #include <vector>
18852 #include <set>
18853 #include <map>
18854 #include <unordered_set>
18855 #include <unordered_map>
18856 #include <queue>
18857 #include <ctime>
18858 #include <cassert>
18859 #include <complex>
18860 #include <string>
18861 #include <cstring>
18862 #include <chrono>
18863 #include <random>
18864 #include <bitset>
18865 #include <array>
18866 #define ll long long int
18867 #define pb push_back
18868 #define mp make_pair
18869 #define mod 1000000007
18870 #define vl vector<ll>
18871 #define all(c) (c).begin(), (c).end()
18872 using namespace std;
18873 ll power(ll a, ll b) {ll res=1; a%=mod; assert(b>=0); for(; b>=1; b--){if(b&1) res=res*a%mod; a
=a*a%mod;} return res;}
18874 ll modInverse(ll a){return power(a, mod-2);}
18875 const int N=500023;
18876 bool vis[N];
18877 vector<int> adj[N];
18878 long long readInt(long long l, long long r, char endd){
18879     long long x=0;
18880     int cnt=0;
18881     int fi=-1;
18882     bool is_neg=false;
18883     while(true){
18884         char g=getchar();
18885         if(g=='-'){
18886             assert(fi==-1);
18887             is_neg=true;
18888             continue;
18889         }
18890         if('0'<=g && g<='9'){
18891             x*=10;
18892             x+=g-'0';
18893             if(cnt==0){
18894                 fi=g-'0';
18895             }
18896             cnt++;
18897             assert(fi!=0 || cnt==1);
18898             assert(fi!=0 || is_neg==false);
18899
18900             assert(!(cnt>19 || ( cnt==19 && fi>1) ));
18901         } else if(g==endd){
18902             if(is_neg){
18903                 x= -x;
18904             }
18905
18906             if(!(l <= x && x <= r))
18907             {
18908                 cerr << l << ' ' << r << ' ' << x << '\n';
18909                 assert(1 == 0);
18910             }
18911
18912             return x;
18913         } else {
18914             assert(false);
18915         }

```

```

18916     }
18917 }
18918 string readString(int l,int r,char endd){
18919     string ret="";
18920     int cnt=0;
18921     while(true){
18922         char g=getchar();
18923         assert(g!=-1);
18924         if(g==endd){
18925             break;
18926         }
18927         cnt++;
18928         ret+=g;
18929     }
18930     assert(l<=cnt && cnt<=r);
18931     return ret;
18932 }
18933 long long readIntSp(long long l,long long r){
18934     return readInt(l,r,' ');
18935 }
18936 long long readIntLn(long long l,long long r){
18937     return readInt(l,r,'\n');
18938 }
18939 string readStringLn(int l,int r){
18940     return readString(l,r,'\n');
18941 }
18942 string readStringSp(int l,int r){
18943     return readString(l,r,' ');
18944 }
18945 int sumN=0;
18946 void solve()
18947 {
18948     int n=readInt(4,100000,'\n');
18949     sumN+=n;
18950     assert(sumN<=200000);
18951     int A[n+1];
18952     int one=0,two=0;
18953     for(int i=1;i<=n;i++)
18954     {
18955         if(i==n)
18956             A[i]=readInt(1,100000,'\n');
18957         else
18958             A[i]=readInt(1,100000,' ');
18959         A[i]%=3;
18960         if(A[i]==1)
18961             one++;
18962         else if(A[i]==2)
18963             two++;
18964     }
18965     int ans=0;
18966     if(one>two)
18967         swap(one,two);
18968     if(one==0 && two==0)
18969     {
18970         cout<<0<<'\n';
18971         return;
18972     }
18973     if(one+two == n)
18974     {
18975         one--;
18976         two--;
18977         ans++;
18978     }
18979     if(one==0 && two==1)
18980     {
18981         ans+=2;
18982         two=3;
18983     }
18984     else if(one==0 && two==2)

```



```

18985     {
18986         ans++;
18987         two=3;
18988     }
18989     int diff=abs(one-two);
18990     while(true)
18991     {
18992         if(diff<=2)
18993             break;
18994         ans++;
18995         diff-=4;
18996     }
18997     diff=abs(diff);
18998     ans+=diff;
18999     int cnt=one+two+diff;
19000     ans+=(cnt/2);
19001     cout<<ans<<'\n';
19002 }
19003 int main()
19004 {
19005     #ifndef ONLINE_JUDGE
19006     freopen("input.txt", "r", stdin);
19007     freopen("output.txt", "w", stdout);
19008     #endif
19009     ios_base::sync_with_stdio(false);
19010     cin.tie(NULL),cout.tie(NULL);
19011     int T=readInt(1,1000,'\n');
19012     while(T--)
19013         solve();
19014     assert(getchar()==-1);
19015     cerr << "Time : " << 1000 * ((double)clock()) / (double)CLOCKS_PER_SEC << "ms\n";
19016 }
19017
19018 //PASSTHRU
19019 #include "bits/stdc++.h"
19020 // #pragma GCC optimize("O3,unroll-loops")
19021 // #pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
19022 using namespace std;
19023 using ll = long long int;
19024 mt19937_64 rng(chrono::high_resolution_clock::now().time_since_epoch().count());
19025
19026 template<class T>
19027 struct RMQ {
19028     vector<vector<T>>> jmp;
19029     RMQ(const vector<T>& V) : jmp(1, V) {
19030         for (int pw = 1, k = 1; pw * 2 <= (int)size(V); pw *= 2, ++k) {
19031             jmp.emplace_back(size(V) - pw * 2 + 1);
19032             for (int j = 0; j < (int)size(jmp[k]); ++j)
19033                 jmp[k][j] = min(jmp[k - 1][j], jmp[k - 1][j + pw]);
19034         }
19035     }
19036     T query(int a, int b) {
19037         assert(a < b); // or return inf if a == b
19038         int dep = 31 - __builtin_clz(b - a);
19039         return min(jmp[dep][a], jmp[dep][b - (1 << dep)]);
19040     }
19041 };
19042
19043 struct LCA {
19044     int T = 0;
19045     vector<int> time, out, dep, path, ret;
19046     RMQ<int> rmq;
19047
19048     LCA(vector<vector<int>>& C) : time(size(C)), out(size(C)), dep(size(C)), rmq((dfs(C,
19049         0,-1), ret)) {}
19050     void dfs(vector<vector<int>>& C, int v, int par) {
19051         time[v] = T++;
19052         for (int y : C[v]) if (y != par) {
19053             path.push_back(v), ret.push_back(time[v]);

```

```

19053         dep[y] = 1 + dep[v];
19054         dfs(C, y, v);
19055     }
19056     out[v] = T;
19057 }
19058
19059 int lca(int a, int b) {
19060     if (a == b) return a;
19061     tie(a, b) = minmax(time[a], time[b]);
19062     return path[rmq.query(a, b)];
19063 }
19064 };
19065
19066 int main()
19067 {
19068     ios::sync_with_stdio(false); cin.tie(0);
19069
19070     int t; cin >> t;
19071     while (t--) {
19072         int n; cin >> n;
19073         vector<int> col(n);
19074         vector adj(n, vector<int>());
19075         vector<array<int, 2>> edges;
19076         map<int, vector<int>> vertices;
19077         for (int i = 0; i < n; ++i) {
19078             cin >> col[i];
19079             vertices[col[i]].push_back(i);
19080         }
19081         for (int i = 0; i < n-1; ++i) {
19082             int u, v; cin >> u >> v;
19083             adj[--u].push_back(--v);
19084             adj[v].push_back(u);
19085             edges.push_back({u, v});
19086         }
19087         LCA L(adj);
19088         vector<ll> ans(n);
19089         auto upd = [&] (int x, int y, int c) { // Add c to the (x, y) path, where x is
an ancestor of y
19090             ans[y] += c;
19091             ans[x] -= c;
19092         };
19093
19094         for (auto &[c, vlist] : vertices) {
19095             // Build virtual tree of vertices with color c, adding to appropriate paths
along the way
19096             sort(begin(vlist), end(vlist), [&] (int u, int v) {return L.time[u] < L.time
[v]});
19097             int k = size(vlist);
19098             for (int i = 0; i+1 < k; ++i) vlist.push_back(L.lca(vlist[i], vlist[i+1]));
19099             sort(begin(vlist), end(vlist), [&] (int u, int v) {return L.time[u] < L.time
[v]});
19100             vlist.erase(unique(begin(vlist), end(vlist)), end(vlist));
19101             stack<int> st;
19102             for (int x : vlist) {
19103                 while (!st.empty()) {
19104                     int u = st.top();
19105                     if (L.out[u] >= L.out[x] and u != x) break;
19106                     st.pop();
19107                 }
19108                 if (!st.empty()) {
19109                     int u = st.top(); // u is the parent of x in this virtual tree
19110                     upd(u, x, c);
19111                 }
19112                 st.push(x);
19113             }
19114         }
19115
19116         auto dfs = [&] (const auto &self, int u, int p) -> void {
19117             for (int v : adj[u]) {

```

```

19118         if (v == p) continue;
19119         self(self, v, u);
19120         ans[u] += ans[v];
19121     }
19122 };
19123 dfs(dfs, 0, 0);
19124 for (auto [u, v] : edges) {
19125     if (L.time[u] > L.time[v]) swap(u, v);
19126     cout << ans[v] << ' ';
19127 }
19128 cout << '\n';
19129 }
19130 }
19131
19132 //ADJPAIRSWAP
19133 //Utkarsh.25dec
19134 #include <bits/stdc++.h>
19135 #define ll long long int
19136 #define pb push_back
19137 #define mp make_pair
19138 #define mod 1000000007
19139 #define vl vector<ll>
19140 #define all(c) (c).begin(), (c).end()
19141 using namespace std;
19142 ll power(ll a, ll b) {ll res=1; a%=mod; assert(b>=0); for(;b;b>>=1){if(b&1) res=res*a%mod; a
=a*a%mod;}return res;}
19143 ll modInverse(ll a){return power(a,mod-2);}
19144 const int N=500023;
19145 bool vis[N];
19146 vector<int> adj[N];
19147 int sumN=0;
19148 vector<pair<int,int>> opers;
19149 int A[N];
19150
19151 // Applying operation on (i,j)
19152 void apply(int i,int j)
19153 {
19154     opers.pb(mp(i,j));
19155     swap(A[i],A[j]);
19156     swap(A[i+1],A[j+1]);
19157 }
19158
19159 // Moving all zeros to left one by one
19160 void placeZeros(int n)
19161 {
19162     // keep will store the position of leftmost 1
19163     int keep=0;
19164     for(int i=2;i<=n;i++)
19165     {
19166         if(A[i]==1 || A[i-1]==0)
19167             continue;
19168         if(A[i-1]==1)
19169         {
19170             while(A[keep]==0)
19171                 keep++;
19172
19173             if(i==n) // Move left once and then move it to required place
19174             {
19175                 apply(n-3,n-1);
19176                 apply(keep,n-2);
19177                 continue;
19178             }
19179
19180             if(keep+1<i) // Directly move it to required place
19181                 apply(keep,i);
19182             else // Move right once and then move it to required place
19183             {
19184                 apply(i,i+2);
19185                 apply(keep,i+2);

```

```

19186     }
19187     }
19188 }
19189 }
19190 void solve()
19191 {
19192    opers.clear();
19193     int n;
19194     cin>>n;
19195     int cnt0=0,cnt1=0;
19196     for(int i=1;i<=n;i++)
19197     {
19198         cin>>A[i];
19199         if(A[i]==0)
19200             cnt0++;
19201         else
19202             cnt1++;
19203     }
19204
19205     // All Same. Already Sorted
19206     if(max(cnt0,cnt1)==n)
19207     {
19208         cout<<0<<"\n";
19209         return;
19210     }
19211
19212     // We will try to place the character with less count
19213     if(cnt0<=cnt1)
19214         placeZeros(n);
19215     else
19216     {
19217         // Reversing and Complimenting the array
19218         int B[n+1];
19219         for(int i=1;i<=n;i++)
19220             B[i]=1-A[n+1-i];
19221         for(int i=1;i<=n;i++)
19222             A[i]=B[i];
19223
19224         placeZeros(n);
19225         // Adjusting the operations as we had reversed the array
19226         for(int i=0;i<opers.size();i++)
19227             opers[i]=mp(n-opers[i].second,n-opers[i].first);
19228     }
19229
19230     cout<<opers.size()<<"\n";
19231     for(auto it:opers)
19232         cout<<it.first<<" "<<it.second<<"\n";
19233 }
19234 int main()
19235 {
19236     ios_base::sync_with_stdio(false);
19237     cin.tie(NULL),cout.tie(NULL);
19238     int T;
19239     cin>>T;
19240     while(T--)
19241         solve();
19242 }
19243
19244 //DISTINCTSEQ
19245 // #pragma GCC optimize("O3")
19246 // #pragma GCC optimize("Ofast,unroll-loops")
19247
19248 #include <bits/stdc++.h>
19249 #include <ext/pb_ds/tree_policy.hpp>
19250 #include <ext/pb_ds/assoc_container.hpp>
19251 using namespace __gnu_pbds;
19252 using namespace std;
19253 #define ll long long
19254 const ll INF_MUL=1e13;

```

```

19255  const ll INF_ADD=1e18;
19256  #define pb push_back
19257  #define mp make_pair
19258  #define nline "\n"
19259  #define f first
19260  #define s second
19261  #define pll pair<ll,ll>
19262  #define all(x) x.begin(),x.end()
19263  #define vl vector<ll>
19264  #define vvl vector<vector<ll>>
19265  #define vvvl vector<vector<vector<ll>>>
19266  #ifndef ONLINE_JUDGE
19267  #define debug(x) cerr<<#x<<" "; _print(x); cerr<<nline;
19268  #else
19269  #define debug(x);
19270  #endif
19271  void _print(ll x){cerr<<x;}
19272  void _print(char x){cerr<<x;}
19273  void _print(string x){cerr<<x;}
19274  mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());
19275  template<class T,class V> void _print(pair<T,V> p) {cerr<<"{"; _print(p.first);cerr<<" ";
; _print(p.second);cerr<<"}";}
19276  template<class T>void _print(vector<T> v) {cerr<<" [ "; for (T i:v){_print(i);cerr<<" "
; }cerr<<"]";}
19277  template<class T>void _print(set<T> v) {cerr<<" [ "; for (T i:v){_print(i); cerr<<" ";}
cerr<<"]";}
19278  template<class T>void _print(multiset<T> v) {cerr<<" [ "; for (T i:v){_print(i);cerr<<"
" ";}cerr<<"]";}
19279  template<class T,class V>void _print(map<T, V> v) {cerr<<" [ "; for(auto i:v) {_print(i
);cerr<<" ";} cerr<<"]";}
19280  typedef tree<ll, null_type, less<ll>, rb_tree_tag, tree_order_statistics_node_update>
ordered_set;
19281  typedef tree<ll, null_type, less_equal<ll>, rb_tree_tag,
tree_order_statistics_node_update> ordered_multiset;
19282  typedef tree<pair<ll,ll>, null_type, less<pair<ll,ll>>, rb_tree_tag,
tree_order_statistics_node_update> ordered_pset;
19283  //-----
-----

19284  const ll MOD=998244353;
19285  const ll MAX=100100;
19286  void solve(){
19287      ll n; cin>>n;
19288      string s; cin>>s;
19289      vector<ll> track[5];
19290      for(ll i=0;i<2*n;i++){
19291          track[s[i]-'0'].push_back(i);
19292      }
19293      if(track[0].size()>track[1].size()){
19294          swap(track[0],track[1]);
19295      }
19296      if(track[0].empty()){
19297          cout<<"-1\n";
19298      }
19299      else{
19300          while(track[0].size()<track[1].size()){
19301              auto it=track[1].back();
19302              track[1].pop_back();
19303              track[0].push_back(it);
19304          }
19305          sort(all(track[0]));
19306          for(auto it:track[0]){
19307              cout<<it+1<<" ";
19308          }
19309          cout<<nline;
19310      }
19311      return;
19312  }
19313  int main()

```

```

19314 {
19315     ios_base::sync_with_stdio(false);
19316     cin.tie(NULL);
19317     #ifndef ONLINE_JUDGE
19318         freopen("input.txt", "r", stdin);
19319         freopen("output.txt", "w", stdout);
19320         freopen("error.txt", "w", stderr);
19321     #endif
19322     ll test_cases=1;
19323     cin>>test_cases;
19324     while(test_cases--){
19325         solve();
19326     }
19327     cout<<fixed<<setprecision(10);
19328     cerr<<"Time:"<<1000*((double)clock())/((double)CLOCKS_PER_SEC<<"ms\n";
19329 }
19330
19331 //SORTPERM
19332 // #pragma GCC optimize("O3")
19333 // #pragma GCC optimize("Ofast,unroll-loops")
19334
19335 #include <bits/stdc++.h>
19336 #include <ext/pb_ds/tree_policy.hpp>
19337 #include <ext/pb_ds/assoc_container.hpp>
19338 using namespace __gnu_pbds;
19339 using namespace std;
19340 #define ll long long
19341 const ll INF_MUL=1e13;
19342 const ll INF_ADD=1e18;
19343 #define pb push_back
19344 #define mp make_pair
19345 #define nline "\n"
19346 #define f first
19347 #define s second
19348 #define pll pair<ll,ll>
19349 #define all(x) x.begin(),x.end()
19350 #define vl vector<ll>
19351 #define vvl vector<vector<ll>>
19352 #define vvvl vector<vector<vector<ll>>>
19353 #ifndef ONLINE_JUDGE
19354 #define debug(x) cerr<<#x<<" "; _print(x); cerr<<nline;
19355 #else
19356 #define debug(x);
19357 #endif
19358 void _print(ll x){cerr<<x;}
19359 void _print(char x){cerr<<x;}
19360 void _print(string x){cerr<<x;}
19361 mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());
19362 template<class T,class V> void _print(pair<T,V> p){cerr<<"{"; _print(p.first);cerr<<" ";
; _print(p.second);cerr<<"}";}
19363 template<class T>void _print(vector<T> v){cerr<<" ["; for (T i:v){_print(i);cerr<<" ";
;}cerr<<" "];}
19364 template<class T>void _print(set<T> v){cerr<<" ["; for (T i:v){_print(i); cerr<<" ";}
cerr<<" "];}
19365 template<class T>void _print(multiset<T> v){cerr<<" ["; for (T i:v){_print(i);cerr<<" ";}
cerr<<" "];}
19366 template<class T,class V>void _print(map<T, V> v){cerr<<" ["; for(auto i:v){_print(i
);cerr<<" ";} cerr<<" "];}
19367 typedef tree<ll, null_type, less<ll>, rb_tree_tag, tree_order_statistics_node_update>
ordered_set;
19368 typedef tree<ll, null_type, less_equal<ll>, rb_tree_tag,
tree_order_statistics_node_update> ordered_multiset;
19369 typedef tree<pair<ll,ll>, null_type, less<pair<ll,ll>>, rb_tree_tag,
tree_order_statistics_node_update> ordered_pset;
19370 //-----
-----
-----

```

```

19371     const ll MOD=998244353;
19372     const ll MAX=500500;
19373     void solve() {
19374         ll n; cin>>n;
19375         vector<ll> p(n+5,0);
19376         ll ans=0;
19377         for(ll i=1;i<=n;i++){
19378             cin>>p[i];
19379             ans+=max(0LL,p[i]-i);
19380         }
19381         cout<<ans<<endl;
19382         vector<pair<ll,ll>> track;
19383         for(ll i=1;i<=n;i++){
19384             while(p[i]!=i){
19385                 ll cur=i;
19386                 for(ll j=i+1;j<=n;j++){
19387                     if(p[j]==i){
19388                         swap(p[j],p[cur]);
19389                         ans+=j-cur;
19390                         track.push_back({j,cur});
19391                         break;
19392                     }
19393                     if(p[j]>p[cur]){
19394                         cur=j;
19395                     }
19396                 }
19397             }
19398         }
19399         cout<<track.size()<<endl;
19400         for(auto it:track){
19401             cout<<it.f<<" "<<it.s<<endl;
19402         }
19403     }
19404     int main()

19405 {
19406     ios_base::sync_with_stdio(false);
19407     cin.tie(NULL);
19408     #ifndef ONLINE_JUDGE
19409     freopen("input.txt", "r", stdin);
19410     freopen("output.txt", "w", stdout);
19411     freopen("error.txt", "w", stderr);
19412     #endif
19413     ll test_cases=1;
19414     cin>>test_cases;
19415     while(test_cases--){
19416         solve();
19417     }
19418     cout<<fixed<<setprecision(10);
19419     cerr<<"Time:"<<1000*((double)clock())/((double)CLOCKS_PER_SEC<<"ms\n";
19420 }
19421
19422 //CHEFDINE
19423 #include <bits/stdc++.h>
19424 using namespace std;
19425
19426
19427 int main(){
19428     // setIO("input");
19429     ios_base::sync_with_stdio(false);
19430     cin.tie();cout.tie();
19431     long long TT = 1;
19432     long long suma = 0;
19433     cin>>TT;
19434     for(long long TR = 1;TR <= TT;TR++){
19435         long long n,k;
19436         cin>>n>>k;
19437         suma+=n;

```

```

19438     assert(n>=1 and n<=100000);
19439     assert(k>=1 and k<=100000);
19440     vector<long long>a(n),b(n);
19441     for(long long i=0;i<n;i++){
19442         cin>>a[i];
19443         assert(a[i]>=1 and a[i]<=100000);
19444     }
19445     for(long long i=0;i<n;i++){
19446         cin>>b[i];
19447         assert(b[i]>=0 and b[i]<=100000);
19448     }
19449     map<long long,long long>mp;
19450     for(long long i=0;i<n;i++){
19451         mp[a[i]] = 10000007;
19452     }
19453     for(long long i=0;i<n;i++){
19454         mp[a[i]] = min(mp[a[i]],b[i]);
19455     }
19456     vector<long long>time;
19457     for(auto it:mp){
19458         if(it.second!=10000007){
19459             time.push_back(it.second);
19460         }
19461     }
19462     sort(time.begin(),time.end());
19463     if(time.size()<k){
19464         cout<<-1<<"\n";
19465         continue;
19466     }
19467     long long ans = 0;
19468     for(long long i=0;i<k;i++){
19469         ans+=time[i];
19470     }
19471     cout<<ans<<"\n";
19472 }
19473 assert(suma>=1 and suma<=100000);
19474 return 0;
19475 }
19476
19477 //RESTORE_
19478 #include <map>
19479 #include <set>
19480 #include <cmath>
19481 #include <ctime>
19482 #include <queue>
19483 #include <stack>
19484 #include <cstdio>
19485 #include <cstdlib>
19486 #include <vector>
19487 #include <cstring>
19488 #include <algorithm>
19489 using namespace std;
19490 typedef double db;
19491 typedef long long ll;
19492 typedef unsigned long long ull;
19493 const int N=1000010;
19494 const int LOGN=28;
19495 const ll TMD=0;
19496 const ll INF=2147483647;
19497 int T,n;
19498 int a[N],ind[N],ans[N];
19499 vector<int> G[N];
19500
19501 int check()
19502 {
19503     for(int i=1;i<=n;i++) if(a[i]>=i) return 1;
19504     return 0;
19505 }
19506

```



```

19507 void topo_sort()
19508 {
19509     int cnt=0;
19510     priority_queue<int,vector<int>,greater<int> > Q;
19511     for(int i=1;i<=n;i++) if(!ind[i]) Q.push(i);
19512     while(!Q.empty())
19513     {
19514         int x=Q.top();
19515         Q.pop();
19516         ans[x]++;cnt;
19517         for(int i=0;i<G[x].size();i++)
19518         {
19519             int y=G[x][i];
19520             ind[y]--;
19521             if(y&&!ind[y]) Q.push(y);
19522         }
19523     }
19524     if(cnt!=n) printf("-1\n");
19525     else for(int i=1;i<=n;i++) printf("%d%c",ans[i],i==n?'\\n':' ');
19526 }
19527
19528 void solve()
19529 {
19530     priority_queue<int> Q;
19531     for(int i=1;i<=n;i++) ind[i]=0,G[i].clear();
19532     for(int i=1;i<=n;i++)
19533     {
19534         if(a[i]==-1) continue;
19535         if(a[i])
19536         {
19537             G[i].push_back(a[i]);
19538             ind[a[i]]++;
19539         }
19540         while(!Q.empty())
19541         {
19542             int t=Q.top();
19543             if(t<=a[i]) break;
19544             Q.pop();
19545             G[t].push_back(i);
19546             ind[i]++;
19547         }
19548         Q.push(i);
19549     }
19550     topo_sort();
19551 }
19552
19553 int main()
19554 {
19555     scanf("%d",&T);
19556     while(T--)
19557     {
19558         scanf("%d",&n);
19559         for(int i=1;i<=n;i++) scanf("%d",&a[i]);
19560         if(check())
19561         {
19562             printf("-1\n");
19563             continue;
19564         }
19565         solve();
19566     }
19567
19568     return 0;
19569 }
19570
19571 //GUESSALL
19572 #include <bits/stdc++.h>
19573 #include <ext/pb_ds/tree_policy.hpp>
19574 #include <ext/pb_ds/assoc_container.hpp>
19575 using namespace __gnu_pbds;

```

```

19576 using namespace std;
19577 #define ll long long
19578 const ll INF_MUL=1e13;
19579 const ll INF_ADD=1e18;
19580 #define pb push_back
19581 #define mp make_pair
19582 #define nline "\n"
19583 #define f first
19584 #define s second
19585 #define pll pair<ll,ll>
19586 #define all(x) x.begin(),x.end()
19587 #define vl vector<ll>
19588 #define vvl vector<vector<ll>>
19589 #define vvvl vector<vector<vector<ll>>>
19590 #ifndef ONLINE_JUDGE
19591 #define debug(x) cerr<<#x<<" "; _print(x); cerr<<nline;
19592 #else
19593 #define debug(x);
19594 #endif
19595 void _print(ll x){cerr<<x;}
19596 void _print(char x){cerr<<x;}
19597 void _print(string x){cerr<<x;}
19598 mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());
19599 template<class T,class V> void _print(pair<T,V> p) {cerr<<"{"; _print(p.first);cerr<<" ";
; _print(p.second);cerr<<"}";}
19600 template<class T>void _print(vector<T> v) {cerr<<" [ "; for (T i:v){_print(i);cerr<<" "
; }cerr<<" ]";}
19601 template<class T>void _print(set<T> v) {cerr<<" [ "; for (T i:v){_print(i); cerr<<" ";}
cerr<<" ]";}
19602 template<class T>void _print(multiset<T> v) {cerr<<" [ "; for (T i:v){_print(i);cerr<<"
 ";}cerr<<" ]";}
19603 template<class T,class V>void _print(map<T, V> v) {cerr<<" [ "; for(auto i:v) {_print(i
);cerr<<" ";} cerr<<" ]";}
19604 typedef tree<ll, null_type, less<ll>, rb_tree_tag, tree_order_statistics_node_update>
ordered_set;
19605 typedef tree<ll, null_type, less_equal<ll>, rb_tree_tag,
tree_order_statistics_node_update> ordered_multiset;
19606 typedef tree<pair<ll,ll>, null_type, less<pair<ll,ll>>, rb_tree_tag,
tree_order_statistics_node_update> ordered_pset;
19607 //-----
-----
19608 const ll MOD=998244353;
19609 const ll MAX=200010;
19610 void solve(){
19611     ll k; cin>>k;
19612     vector<ll> track(k+5,0);
19613     ll n; cin>>n;
19614     vector<ll> b(n+5,0);
19615     set<ll> check;
19616     for(ll i=1;i<=n;i++){
19617         cin>>b[i];
19618         check.insert(b[i]%(k+1));
19619     }
19620     if(check.size()<k){
19621         for(auto it:check){
19622             cout<<"? "<<it<<endl;
19623             cin>>track[it];
19624         }
19625         cout<<"! ";
19626         for(ll i=1;i<=n;i++){
19627             cout<<track[b[i]%(k+1)]<<" ";
19628         }
19629         cout<<endl;
19630     }
19631     else{
19632         for(ll i=0;i<k;i++){
19633             cout<<"? "<<i<<endl;
19634             cin>>track[i];

```

```

19635         track[k]-=track[i];
19636     }
19637     cout<<"! ";
19638     for (ll i=1;i<=n;i++){
19639         cout<<track[b[i]%(k+1)]<<" ";
19640     }
19641     cout<<endl;
19642 }
19643 return;
19644 }
19645 int main()

19646 {
19647     ios_base::sync_with_stdio(false);
19648     cin.tie(NULL);
19649     #ifndef ONLINE_JUDGE
19650     freopen("input.txt", "r", stdin);
19651     freopen("output.txt", "w", stdout);
19652     freopen("error.txt", "w", stderr);
19653     #endif
19654     ll test_cases=1;
19655     cin>>test_cases;
19656     while(test_cases--){
19657         solve();
19658     }
19659     cout<<fixed<<setprecision(10);
19660     cerr<<"Time:"<<1000*((double)clock())/(double)CLOCKS_PER_SEC<<"ms\n";
19661 }
19662
19663 //APTREE - BINARY LIFTING
19664 #include <bits/stdc++.h>
19665 using namespace std;
19666 #define ll long long
19667 #define ull unsigned long long
19668 #define pb(e) push_back(e)
19669 #define sv(a) sort(a.begin(),a.end())
19670 #define sa(a,n) sort(a,a+n)
19671 #define mp(a,b) make_pair(a,b)
19672 #define vf first
19673 #define vs second
19674 #define ar array
19675 #define all(x) x.begin(),x.end()
19676 const int inf = 0x3f3f3f3f;
19677 const int mod = 998244353;
19678 const double PI=3.14159265358979323846264338327950288419716939937510582097494459230;
19679 bool remender(ll a , ll b){return a%b;}
19680
19681 //freopen("problemname.in", "r", stdin);
19682 //freopen("problemname.out", "w", stdout);
19683
19684 struct item {
19685     int down , vald , up , valu , vald1 , valu1 , best , full;
19686 };
19687
19688 int isap(int a , int b , int c , int d){
19689     if(d == -1 && a == -1)return 1;
19690     if(a == -1){
19691         if(c - b == d - c)return 1;
19692         return 0;
19693     }
19694     if(d == -1){
19695         if(b - a == c - b)return 1;
19696         return 0;
19697     }
19698     if(b - a == c - b && c - b == d - c)return 1;
19699     int cnt = 0;
19700     if(c - b == b - a)cnt = 2;
19701     if(c - b == d - c){

```

```

19702         if(cnt == 1)cnt = 4;
19703         else cnt = 3;
19704     }
19705     return cnt;
19706 }
19707
19708 const int N = 200003 , L = 22;
19709
19710 vector<int> adj[N];
19711 int arr[N];
19712 int timer , tin[N] , tout[N];
19713 item up[N][L];
19714 int p[N][L];
19715
19716 item merge(item a , item b , int pr = 0){
19717     item ans;
19718     ans.best = max(a.best , b.best);
19719     ans.valu = a.valu;
19720     ans.vald = b.vald;
19721     ans.valu1 = a.valu1;
19722     if(a.valu1 == -1){
19723         ans.valu1 = b.valu;
19724     }
19725     ans.vald1 = b.vald1;
19726     ans.full = 0;
19727     if(b.vald1 == -1){
19728         ans.vald1 = a.vald;
19729     }
19730     ans.up = a.up;
19731     ans.down = b.down;
19732     int x = isap(b.valu1 , b.valu , a.vald , a.vald1);
19733     if(x > 0){
19734         if(x == 1){
19735             ans.best = max(ans.best , a.down + b.up);
19736             if(a.full && b.full){
19737                 ans.full = 1;
19738             }
19739             if(b.full){
19740                 ans.down = a.down + b.up;
19741             }
19742             if(a.full){
19743                 ans.up = a.down + b.up;
19744             }
19745         }
19746         else if(x == 2){
19747             ans.best = max(ans.best , b.up + 1);
19748             if(b.full){
19749                 ans.down++;
19750             }
19751         }
19752         else if(x == 3){
19753             ans.best = max(ans.best , a.down + 1);
19754             if(a.full)ans.up++;
19755         }
19756         else {
19757             ans.best = max({ans.best , a.down + 1 , b.up + 1});
19758             if(b.full){
19759                 ans.down++;
19760             }
19761             if(a.full)ans.up++;
19762         }
19763     }
19764     if(ans.full){
19765         ans.up = ans.down = ans.best;
19766     }
19767     ans.down = max(ans.down , 2);
19768     ans.up = max(ans.up , 2);
19769     return ans;
19770 }

```

```

19771
19772 void dfs(int node , int par , int dis){
19773     tin[node] = timer++;
19774     up[node][0] = {1 , arr[node] , 1 , arr[node] , -1 , -1 , 1 , 1};
19775     p[node][0] = par;
19776     for(int i = 1; i < L; i++){
19777         if(dis < (1 << i)){
19778             up[node][i] = up[node][i-1];
19779             p[node][i] = p[p[node][i-1]][i-1];
19780             continue;
19781         }
19782         up[node][i] = merge(up[p[node][i-1]][i-1] , up[node][i-1]);
19783         p[node][i] = p[p[node][i-1]][i-1];
19784     }
19785     for(int i : adj[node]){
19786         if(i != par){
19787             dfs(i , node , dis + 1);
19788         }
19789     }
19790     tout[node] = timer++;
19791 }
19792
19793 bool islca(int x , int y){
19794     return tin[x] <= tin[y] && tout[x] >= tout[y];
19795 }
19796
19797 int find(int u , int v){
19798     if(islca(u , v))return u;
19799     else if(islca(v , u))return v;
19800     for(int i = L - 1; i >= 0; i--){
19801         if(!islca(p[u][i],v))u = p[u][i];
19802     }
19803     return p[u][0];
19804 }
19805
19806 item corner(int lca , int x , int todo = 0){
19807     item cur = {1 , arr[x] , 1 , arr[x] , -1 , -1 , 1 ,1};
19808     x = p[x][0];
19809     for(int i = L - 1; i >= 0; i--){
19810         if(!islca(p[x][i] , lca)){
19811             cur = merge(up[x][i] , cur);
19812             x = p[x][i];
19813         }
19814     }
19815     if(x != lca){
19816         cur = merge(up[x][0], cur);
19817         x = p[x][0];
19818     }
19819     if(todo == 0)cur = merge(up[x][0], cur);
19820     return cur;
19821 }
19822
19823 void solve(){
19824     int n;
19825     cin >> n;
19826     for(int i = 1; i <= n; i++)cin >> arr[i];
19827     for(int i = 0; i < n-1; i++){
19828         int u , v;
19829         cin >> u >> v;
19830         adj[u].pb(v);
19831         adj[v].pb(u);
19832     }
19833     dfs(1 , 1 , 1);
19834     int q;
19835     cin >> q;
19836     while(q--){
19837         int u , v;
19838         cin >> u >> v;
19839         if(u == v){

```

```

19840         cout << 1 << '\n';
19841         continue;
19842     }
19843     int lca = find(u , v);
19844     if(lca == u){
19845         cout << corner(lca , v).best << '\n';
19846     }
19847     else if(lca == v){
19848         cout << corner(lca , u).best << '\n';
19849     }
19850     else {
19851         item x = corner(lca ,u);
19852         item y = corner(lca, v, 1);
19853         swap(x.valu , x.vald);
19854         swap(x.valu1,x.vald1);
19855         swap(x.up , x.down);
19856         cout << merge(x,y).best << '\n';
19857     }
19858 }
19859 }
19860
19861 int main(){
19862     ios_base::sync_with_stdio(false);
19863     cin.tie(NULL);
19864     //int t;cin >> t;while(t--)
19865     solve();
19866     return 0;
19867 }
19868
19869 //XORPROD
19870 #include <bits/stdc++.h>
19871 #define mod 998244353
19872 using namespace std;
19873
19874 int main() {
19875     //freopen("inp4.in", "r", stdin);
19876     //freopen("out4.out", "w", stdout);
19877     int t;
19878     cin >> t;
19879     assert(t > 0 && t < 50000);
19880     while(t--) {
19881         int n;
19882         cin >> n;
19883         assert(n > 0 && n <= 100000);
19884         long long int a[n];
19885         int ones = 0;
19886         priority_queue<long long int> pq;
19887         long long int ans = 1;
19888         for(int i = 0; i < n; i++) {
19889             cin >> a[i];
19890             assert(a[i] > 0 && a[i] <= 1000000000);
19891             if(a[i]&1) {
19892                 if(a[i] == 1) ones++;
19893                 ans *= a[i];
19894                 ans %= mod;
19895             } else pq.push(-a[i]);
19896         }
19897         while(ones && !pq.empty()) {
19898             int top = -pq.top();
19899             pq.pop();
19900             ones--;
19901             ans *= (top + 1);
19902             ans %= mod;
19903         }
19904         while(!pq.empty()) {
19905             ans *= (-pq.top());
19906             pq.pop();
19907             ans %= mod;
19908         }

```

```

19909         cout << ans << "\n";
19910     }
19911 }
19912
19913 //INTARR
19914 #include<bits/stdc++.h>
19915 using namespace std;
19916
19917 #include <ext/pb_ds/assoc_container.hpp>
19918 #include <ext/pb_ds/tree_policy.hpp>
19919 using namespace __gnu_pbds;
19920
19921 #define ll long long
19922 #define db double
19923 #define el "\n"
19924 #define ld long double
19925 #define rep(i,n) for(int i=0;i<n;i++)
19926 #define rev(i,n) for(int i=n;i>=0;i--)
19927 #define rep_a(i,a,n) for(int i=a;i<n;i++)
19928 #define all(ds) ds.begin(), ds.end()
19929 #define ff first
19930 #define ss second
19931 #define pb push_back
19932 #define mp make_pair
19933 typedef vector< long long > vi;
19934 typedef pair<long long, long long> ii;
19935 typedef priority_queue <ll> pq;
19936 #define o_set tree<ll, null_type,less<ll>,
rb_tree_tag,tree_order_statistics_node_update>
19937
19938 const ll mod = 1000000007;
19939 const ll INF = (ll)1e18;
19940 const ll MAXN = 1000006;
19941
19942 ll po(ll x, ll n){
19943     ll ans=1;
19944     while(n>0){ if(n&1) ans=(ans*x)%mod; x=(x*x)%mod; n/=2;}
19945     return ans;
19946 }
19947
19948 bool fun(vector<ll> &a){
19949     int n = a.size();
19950     int c[n];
19951
19952     int j = (n+1)/2;
19953
19954     c[0] = a[0];
19955     int k = 1;
19956     for(int i = 1; i<(n+1)/2; i++){
19957         c[k++] = a[j++];
19958         c[k++] = a[i];
19959     }
19960     if(k<n) c[k] = a[j];
19961     int ok = 1;
19962
19963     for(int i=1; i+1<n; i++){
19964         ok &= ( !(c[i-1] <= c[i] && c[i] <= c[i+1])
19965                && !(c[i-1] >= c[i] && c[i] >= c[i+1])));
19966     }
19967
19968     if(ok){
19969         rep(i,n) cout<<c[i]<<" ";
19970         cout<<el;
19971         return true;
19972     }
19973
19974     j = n/2;
19975     k = 0;
19976     for(int i = 0; j < n; i++){

```

```

19977         c[k++] = a[j++];
19978         if(i<n/2) c[k++] = a[i];
19979     }
19980
19981     ok = 1;
19982     for(int i=1; i+1<n; i++){
19983         ok &= ( ! (c[i-1] <= c[i] && c[i] <= c[i+1])
19984                && ! (c[i-1] >= c[i] && c[i] >= c[i+1]));
19985     }
19986
19987     if(ok){
19988         rep(i,n) cout<<c[i]<<" ";
19989         cout<<el;
19990         return true;
19991     }
19992     return ok;
19993 }
19994
19995
19996 int main(){
19997     ios_base::sync_with_stdio(false);
19998     cin.tie(0);
19999     cout.tie(0);
20000     #ifndef ONLINE_JUDGE
20001     freopen("input.txt", "r" , stdin);
20002     freopen("output.txt", "w" , stdout);
20003     #endif
20004     int T=1;
20005     cin >> T;
20006     while(T--){
20007         int n;
20008         cin>>n;
20009
20010         vector<ll> a(n);
20011         rep(i,n) cin>>a[i];
20012
20013         sort(all(a));
20014
20015         bool z = fun(a);
20016         if(!z){
20017
20018             cout<<-1<<el;
20019         }
20020
20021     }
20022     cerr << "Time : " << 1000 * ((double)clock()) / (double)CLOCKS_PER_SEC << "ms\n";
20023     return 0;
20024 }
20025
20026
20027 //BORSTR
20028 #include<bits/stdc++.h>
20029 using namespace std;
20030
20031 #include <ext/pb_ds/assoc_container.hpp>
20032 #include <ext/pb_ds/tree_policy.hpp>
20033 using namespace __gnu_pbds;
20034
20035 #define ll long long
20036 #define db double
20037 #define el "\n"
20038 #define ld long double
20039 #define rep(i,n) for(int i=0;i<n;i++)
20040 #define rev(i,n) for(int i=n;i>=0;i--)
20041 #define rep_a(i,a,n) for(int i=a;i<n;i++)
20042 #define all(ds) ds.begin(), ds.end()
20043 #define ff first
20044 #define ss second
20045 #define pb push_back

```



```

20046 #define mp make_pair
20047 typedef vector< long long > vi;
20048 typedef pair<long long, long long> ii;
20049 typedef priority_queue <ll> pq;
20050 #define o_set tree<ll, null_type,less<ll>,
rb_tree_tag,tree_order_statistics_node_update>
20051
20052 const ll mod = 1000000007;
20053 const ll INF = (ll)1e18;
20054 const ll MAXN = 1000006;
20055
20056 ll po(ll x, ll n){
20057     ll ans=1;
20058     while(n>0){ if(n&1) ans=(ans*x)%mod; x=(x*x)%mod; n/=2;}
20059     return ans;
20060 }
20061
20062
20063 int main(){
20064     ios_base::sync_with_stdio(false);
20065     cin.tie(0);
20066     cout.tie(0);
20067
20068     int T=1;
20069     cin >> T;
20070     while(T--){
20071         int n;
20072         cin>>n;
20073
20074         string s;
20075         cin>>s;
20076
20077         assert(s.length()==n);
20078         for(auto h:s){
20079             assert(h>='a' && h<='z');
20080         }
20081
20082         vector<int>len(26, 0);
20083         vector<int>cnt(26, 0);
20084
20085         int curr = 1;
20086
20087         rep_a(i,1,n+1){
20088             if(s[i]!=s[i-1] || i==n+1){
20089                 int id = (int)(s[i-1]-'a');
20090                 if(curr>len[id]){
20091                     len[id]=curr;
20092                     cnt[id]=1;
20093                 }
20094                 else if(curr==len[id]){
20095                     cnt[id]++;
20096                 }
20097                 curr=1;
20098             }
20099             else curr++;
20100         }
20101
20102         int mx = 0, id, ans;
20103         rep(i,26){
20104             if(len[i]>mx){
20105                 mx = len[i];
20106                 if(cnt[i]>1){
20107                     ans = len[i];
20108                 }
20109                 else{
20110                     ans = len[i]-1;
20111                     mx--;
20112                 }
20113             }

```

```

20114     }
20115
20116     cout<<ans<<el;
20117
20118
20119 }
20120 cerr << "Time : " << 1000 * ((double)clock()) / (double)CLOCKS_PER_SEC << "ms\n";
20121 return 0;
20122 }
20123
20124 //CNTNOPARS
20125 #include <bits/stdc++.h>
20126 #include <ext/pb_ds/tree_policy.hpp>
20127 #include <ext/pb_ds/assoc_container.hpp>
20128 using namespace __gnu_pbds;
20129 using namespace std;
20130 #define ll long long
20131 const ll INF_MUL=1e13;
20132 const ll INF_ADD=1e18;
20133 #define pb push_back
20134 #define mp make_pair
20135 #define nline "\n"
20136 #define f first
20137 #define s second
20138 #define pll pair<ll,ll>
20139 #define all(x) x.begin(),x.end()
20140 #define vl vector<ll>
20141 #define vvl vector<vector<ll>>
20142 #define vvvl vector<vector<vector<ll>>>
20143 #ifndef ONLINE_JUDGE
20144 #define debug(x) cerr<<#x<<" "; _print(x); cerr<<nline;
20145 #else
20146 #define debug(x);
20147 #endif
20148 void _print(ll x){cerr<<x;}
20149 void _print(char x){cerr<<x;}
20150 void _print(string x){cerr<<x;}
20151 mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());
20152 template<class T,class V> void _print(pair<T,V> p) {cerr<<"{"; _print(p.first);cerr<<"",
; _print(p.second);cerr<<"}";}
20153 template<class T>void _print(vector<T> v) {cerr<<" [ "; for (T i:v){_print(i);cerr<<" "
;};cerr<<"]";}
20154 template<class T>void _print(set<T> v) {cerr<<" [ "; for (T i:v){_print(i); cerr<<" ";}
cerr<<"]";}
20155 template<class T>void _print(multiset<T> v) {cerr<<" [ "; for (T i:v){_print(i);cerr<<
" ";};cerr<<"]";}
20156 template<class T,class V>void _print(map<T, V> v) {cerr<<" [ "; for(auto i:v) {_print(i
);cerr<<" "; cerr<<"]";}
20157 typedef tree<ll, null_type, less<ll>, rb_tree_tag, tree_order_statistics_node_update>
ordered_set;
20158 typedef tree<ll, null_type, less_equal<ll>, rb_tree_tag,
tree_order_statistics_node_update> ordered_multiset;
20159 typedef tree<pair<ll,ll>, null_type, less<pair<ll,ll>>, rb_tree_tag,
tree_order_statistics_node_update> ordered_pset;
20160 //-----
-----
20161 const ll MOD=998244353;
20162 const ll MAX=5000500;
20163 vector<ll> freqpos(MAX,0),freqneg(MAX,0);
20164 void solve(){
20165     ll n; cin>>n;
20166     vector<ll> p(n+5,0);
20167     ll ans=0;
20168     for(ll i=1;i<=n;i++){
20169         cin>>p[i];
20170         if(p[i]%i){
20171             ;
20172         }

```

```

20173         else{
20174             ans++;
20175         }
20176     }
20177     for(ll k=1;k<=2*n;k++){
20178         ll till=min(n,(2*n)/k);
20179         ll zero=0;
20180         for(ll i=1;i<=till;i++){
20181             ll now=p[i]-i*k;
20182             assert(abs(now)<MAX);
20183             if(now==0){
20184                 ans+=zero++;
20185             }
20186             else if(now>0){
20187                 ans+=freqneg[now];
20188                 freqpos[now]++;
20189             }
20190             else{
20191                 ans+=freqpos[-now];
20192                 freqneg[-now]++;
20193             }
20194         }
20195         for(ll i=1;i<=till;i++){
20196             ll now=p[i]-i*k;
20197             if(now==0){
20198                 ;
20199             }
20200             else if(now>0){
20201                 freqpos[now]--;
20202             }
20203             else{
20204                 freqneg[-now]--;
20205             }
20206         }
20207     }
20208     cout<<ans<<endl;
20209     return;
20210 }
20211 int main()

20212 {
20213     ios_base::sync_with_stdio(false);
20214     cin.tie(NULL);
20215     #ifndef ONLINE_JUDGE
20216     freopen("input.txt", "r", stdin);
20217     freopen("output.txt", "w", stdout);
20218     freopen("error.txt", "w", stderr);
20219     #endif
20220     ll test_cases=1;
20221     cin>>test_cases;
20222     while(test_cases--){
20223         solve();
20224     }
20225     cout<<fixed<<setprecision(10);
20226     cerr<<"Time:"<<1000*((double)clock())/((double)CLOCKS_PER_SEC)<<"ms\n";
20227 }

20228
20229 //A_S_P
20230 #define ll long long int
20231 #include<bits/stdc++.h>
20232 #define loop(i,a,b) for(ll i=a;i<b;++i)
20233 #define rloop(i,a,b) for(ll i=a;i>=b;i--)
20234 #define in(a,n) for(ll i=0;i<n;++i) cin>>a[i];
20235 #define pb push_back
20236 #define mk make_pair
20237 #define all(v) v.begin(),v.end()
20238 #define dis(v) for(auto i:v)cout<<i<<" ";cout<<endl;
20239 #define display(arr,n) for(int i=0;i<n;i++)cout<<arr[i]<<" ";cout<<endl;

```

```

20240 #define fast
ios_base::sync_with_stdio(false);cin.tie(NULL);cout.tie(NULL);srand(time(NULL));
20241 #define l(a) a.length()
20242 #define fr first
20243 #define sc second
20244 #define mod 1000000007
20245 #define endl '\n'
20246 #define yes cout<<"Yes"<<endl;
20247 #define no cout<<"No"<<endl;
20248 using namespace std;
20249 #define debug(x) cerr << #x<<" "; _print(x); cerr << endl;
20250 void _print(ll t) {cerr << t;}
20251 void _print(int t) {cerr << t;}
20252 void _print(string t) {cerr << t;}
20253 void _print(char t) {cerr << t;}
20254 void _print(double t) {cerr << t;}
20255 template <class T, class V> void _print(pair <T, V> p);
20256 template <class T> void _print(vector <T> v);
20257 template <class T> void _print(set <T> v);
20258 template <class T, class V> void _print(map <T, V> v);
20259 template <class T> void _print(multiset <T> v);
20260 template <class T, class V> void _print(pair <T, V> p) {cerr << "{"; _print(p.fr); cerr
<< ", "; _print(p.sc); cerr << "}";}
20261 template <class T> void _print(vector <T> v) {cerr << "["; for (T i : v) {_print(i);
cerr << " ";} cerr << "];"}
20262 template <class T> void _print(set <T> v) {cerr << "["; for (T i : v) {_print(i); cerr
<< " ";} cerr << "];"}
20263 template <class T> void _print(multiset <T> v) {cerr << "["; for (T i : v) {_print(i);
cerr << " ";} cerr << "];"}
20264 template <class T, class V> void _print(map <T, V> v) {cerr << "["; for (auto i : v) {
_print(i); cerr << " ";} cerr << "];"}
20265
20266 ll add(ll x,ll y) {ll ans = x+y; return (ans>=mod ? ans - mod : ans);}
20267 ll sub(ll x,ll y) {ll ans = x-y; return (ans<0 ? ans + mod : ans);}
20268 ll mul(ll x,ll y) {ll ans = x*y; return (ans>=mod ? ans % mod : ans);}
20269
20270
20271 #define level 20
20272 vector<map<int,int>> t;
20273 vector<map<int,int>> old,newly_added;
20274
20275 void build(vector<int> &a, int v, int tl, int tr) {
20276     if (tl == tr) {
20277         map<int,int> mp; mp[a[tl]]++;
20278         t[v] = mp;
20279     } else {
20280         int tm = (tl + tr) / 2;
20281         build(a, v*2, tl, tm);
20282         build(a, v*2+1, tm+1, tr);
20283         for(auto i:t[2*v]) t[v][i.fr]+=i.sc;
20284         for(auto i:t[2*v+1]) t[v][i.fr]+=i.sc;
20285     }
20286 }
20287
20288
20289
20290
20291 int query(int v, int tl, int tr, int l, int r, int x) {
20292     if (l > r)
20293         return 0;
20294     if(old[v].size()){ // if it has already been updated previously.
20295         if(tl!=tr) old[2*v] = old[2*v+1] = old[v];
20296         t[v] = old[v];
20297         t[v].begin()->second = tr-tl+1;
20298         old[v].clear();
20299     }
20300     if (l == tl && r == tr) {
20301         if(t[v].find(x) == t[v].end()) return 0;
20302         return t[v][x];

```

```

20303     }
20304     int tm = (tl + tr) / 2;
20305     return (query(v*2, tl, tm, l, min(r, tm), x) + query(v*2+1, tm+1, tr, max(l, tm+1),
    r, x));
20306 }
20307
20308
20309
20310
20311 void update(int v, int tl, int tr, int l,int r, int new_val) {
20312     if(l > r)     return;
20313     // here if in some previous query we updated range from 1 to 4 but now in another
    query if want to break that 1 to 4 than we have to update its children so for that
    reason I created old vector in that i will store the previous value.
20314     if(old[v].size()){ // if it has already been updated previously.
20315         if(tl!=tr)    old[2*v] = old[2*v+1] = old[v];
20316         t[v] = old[v];
20317         t[v].begin()->second = tr-tl+1;
20318         old[v].clear();
20319     }
20320     // if it is in the updation range then I have to remove those color from its
    parents as well so i store those color in newly_added vector.
20321     if(l == tl && r == tr){
20322         newly_added[v] = t[v];
20323         newly_added[v][new_val]--(tr-tl+1);
20324         t[v].clear();
20325         t[v][new_val]+=(tr-tl+1);
20326         // for its children I am storing its current value.
20327         if(tl!=tr)    old[2*v] = old[2*v+1] = t[v];
20328     }
20329     else{
20330         int tm = (tl+tr)/2;
20331         update(2*v,tl,tm,l,min(r,tm),new_val);
20332         update(2*v+1,tm+1,tr,max(l,tm+1),r,new_val);
20333         // now if its children have made some changes in his vector then we are storing
        it in newly_added vector so from that we have to erase those color as well from
        the node u.
20334         if(tl!=tr && newly_added[2*v].size()){
20335             if(v!=1){
20336                 newly_added[v] = newly_added[2*v];
20337             }
20338             set<int> zero_count;
20339             for(auto i:newly_added[2*v]) {
20340                 t[v][i.fr]--i.sc;
20341                 if(t[v][i.fr] == 0) zero_count.insert(i.fr);
20342             }
20343             for(auto i:zero_count) t[v].erase(i);
20344             newly_added[2*v].clear();
20345         }
20346         if(tl!=tr && newly_added[2*v+1].size()){
20347             if(v!=1){
20348                 for(auto i:newly_added[2*v+1])    newly_added[v][i.fr]+=i.sc;
20349             }
20350             set<int> zero_count;
20351             for(auto i:newly_added[2*v+1]) {
20352                 t[v][i.fr]--i.sc;
20353                 if(t[v][i.fr] == 0) zero_count.insert(i.fr);
20354             }
20355             for(auto i:zero_count) t[v].erase(i);
20356             newly_added[2*v+1].clear();
20357         }
20358     }
20359 }
20360
20361 vector<vector<int>> vec;
20362 vector<vector<int>> parent;
20363 vector<int> col,in,out;
20364 vector<int> v,height;
20365

```

```

20366
20367 void precomputeSparseMatrix(int n)
20368 {
20369     for (int i=1; i<level; i++)
20370         for (int node = 1; node <= n; node++)
20371             parent[node][i] = parent[parent[node][i-1]][i-1];
20372 }
20373
20374
20375 int binary_lift(ll vl, ll curr_node){
20376     loop(i, 0, level)
20377         if(vl & (1ll << i))
20378             curr_node = parent[curr_node][i];
20379     return curr_node;
20380 }
20381
20382
20383
20384
20385 void dfs(int i, int par, int &time, int h){
20386     // cerr<<i<<" ";
20387     in[i] = ++time;
20388     height[i] = h;
20389     parent[i][0] = par;
20390     v.push_back(col[i-1]);
20391     for(auto j:vec[i]) {
20392         if(j!=par)
20393             dfs(j, i, time, h+1);
20394     }
20395     out[i] = ++time;
20396     v.pb(col[i-1]);
20397     // cerr<<i<<" ";
20398 }
20399
20400 ll cnt = 0, tot1 = 0, tot2 = 0;
20401
20402 void solve(){
20403     int n; cin>>n;
20404     assert(n>=1 && n<=3e5);
20405     col.assign(n, 0); in(col, n);
20406     loop(i, 0, n) assert(col[i]>=1 && col[i]<=1e9);
20407     tot1+=n;
20408     in.assign(n+1, 0);
20409     out.assign(n+1, 0);
20410     vec.assign(n+1, {});
20411     height.assign(n+1, 0);
20412     v.clear();
20413     loop(i, 0, n-1){
20414         int a, b; cin>>a>>b;
20415         assert(a>=1 && a<=n);
20416         assert(b>=1 && b<=n);
20417         vec[a].pb(b);
20418         vec[b].pb(a);
20419     }
20420     int time = 0;
20421     parent.assign(n+1, vector<int>(level, 0));
20422     dfs(1, 0, time, 0);
20423     precomputeSparseMatrix(n);
20424     int sz = v.size();
20425     sz = ceil(1.00*log2(v.size()));
20426     sz = (1ll << sz); sz*=2;
20427     t.assign(sz, {});
20428     old.assign(sz, {});
20429     newly_added.assign(sz, {});
20430     build(v, 1, 0, v.size()-1);
20431     int q; cin>>q;
20432     int curr_root = 1;
20433     loop(i, 0, q){
20434         int type; cin>>type;

```

```

20435 // debug(type)
20436 assert(type >= 1 && type <= 3);
20437 if(type == 1){
20438     int x,col;    cin>>x>>col;
20439     assert(x>=1 && x<=n);
20440     assert(col>=1 && col<=1e9);
20441     if(curr_root != 1 && (in[x] <= in[curr_root] && out[x] >= out[curr_root])){
20442         int diff = height[curr_root]-height[x];
20443         int node = (diff ? binary_lift(diff-1,curr_root) : curr_root);
20444         if(diff == 0){
20445             int l = 1,r = v.size();
20446             update(1,0,v.size()-1,l-1,r-1,col);
20447         }
20448         else{
20449             int l_1 = 1,r_1 = in[node]-1,l_2 = out[node]+1,r_2 = v.size();
20450             update(1,0,v.size()-1,l_1-1,r_1-1,col);
20451             update(1,0,v.size()-1,l_2-1,r_2-1,col);
20452         }
20453     }
20454     else{
20455         int l = in[x],r = out[x];
20456         update(1,0,v.size()-1,l-1,r-1,col);
20457     }
20458 }
20459 else if(type == 2){
20460     int x;    cin>>x;
20461     assert(x>=1 && x<=n);
20462     curr_root = x;
20463 }
20464 else{
20465     int x,u; cin>>x>>u;
20466     assert(x>=1 && x<=n);
20467     assert(u>=1 && u<=1e9);
20468     ll ans = 0;
20469     if(curr_root != 1 && (in[x]<=in[curr_root] && out[x] >= out[curr_root])){
20470         int diff = height[curr_root]-height[x];
20471         int node = (diff ? binary_lift(diff-1,curr_root) : curr_root);
20472         if(diff == 0){
20473             int l = 1,r = v.size();
20474             ans+=query(1,0,v.size()-1,l-1,r-1,u);
20475         }
20476         else{
20477             int l_1 = 1,r_1 = in[node]-1,l_2 = out[node]+1,r_2 = v.size();
20478             ans+=query(1,0,v.size()-1,l_1-1,r_1-1,u);
20479             ans+=query(1,0,v.size()-1,l_2-1,r_2-1,u);
20480         }
20481     }
20482     else{
20483         int l = in[x],r = out[x];
20484         ans+=query(1,0,v.size()-1,l-1,r-1,u);
20485     }
20486     cout<<ans/2<<endl;
20487 }
20488 }
20489 }
20490
20491
20492 int main()
20493 {
20494     fast
20495     int t; cin>>t;
20496     while(t--) solve();
20497     return 0;
20498 }
20499
20500 //A_S_P-EDITOR
20501 #include "bits/stdc++.h"
20502 // #pragma GCC optimize("O3,unroll-loops")
20503 // #pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")

```

```

20504 using namespace std;
20505 using ll = long long int;
20506 mt19937_64 rng(chrono::high_resolution_clock::now().time_since_epoch().count());
20507
20508 #include <bits/extc++.h>
20509 using namespace __gnu_pbds;
20510 struct chash {
20511     const int RANDOM = (int64_t)(make_unique<char>().get()) ^ chrono::
high_resolution_clock::now().time_since_epoch().count();
20512     static uint64_t hash_f(uint64_t x) {
20513         x += 0x9e3779b97f4a7c15;
20514         x = (x ^ (x >> 30)) * 0xbf58476d1ce4e5b9;
20515         x = (x ^ (x >> 27)) * 0x94d049bb133111eb;
20516         return x ^ (x >> 31);
20517     }
20518     static unsigned hash_combine(unsigned a, unsigned b) { return a * 31 + b; }
20519     int operator()(int x) const { return hash_f(x)^RANDOM; }
20520 };
20521 using hashmap = gp_hash_table<int, int, chash>;
20522
20523 struct Node {
20524     hashmap freq;
20525     Node *l = 0, *r = 0, *par = 0;
20526     int lo, hi;
20527     int change = false, val = 0;
20528     Node(int _lo, int _hi) : lo(_lo), hi(_hi) {}
20529     int query(int L, int R, int x) {
20530         if (R <= lo || hi <= L) return 0;
20531         if (L <= lo && hi <= R) {
20532             return freq[x];
20533         }
20534         push();
20535         return l->query(L, R, x) + r->query(L, R, x);
20536     }
20537     void set(int pos, int x) {
20538         freq[x]++;
20539         if (lo+1 == hi) return;
20540         push();
20541         if (pos >= r->lo) r->set(pos, x);
20542         else l->set(pos, x);
20543     }
20544     void rangeset(int L, int R, int x, bool updpar = false) {
20545         if (R <= lo || hi <= L) return;
20546         if (L <= lo && hi <= R) {
20547             change = true;
20548             val = x;
20549             if (updpar) {
20550                 auto cur = par;
20551                 while (cur) {
20552                     for (auto &[val, ct] : freq) {
20553                         cur -> freq[val] -= ct;
20554                         if (cur -> freq[val] == 0) (cur -> freq).erase(val);
20555                     }
20556                     cur -> freq[x] += hi - lo;
20557                     cur = cur -> par;
20558                 }
20559             }
20560             freq.clear();
20561             freq[x] = hi - lo;
20562             return;
20563         }
20564         push();
20565         l->rangeset(L, R, x, updpar);
20566         r->rangeset(L, R, x, updpar);
20567     }
20568     void push() {
20569         if (!l) {
20570             int mid = lo + (hi - lo)/2;
20571             l = new Node(lo, mid); r = new Node(mid, hi);

```



```

20572         l -> par = r -> par = this;
20573     }
20574     if (change)
20575         l->rangeset(lo,hi,val), r->rangeset(lo,hi,val), change = false;
20576 }
20577 };
20578
20579 int main()
20580 {
20581     ios::sync_with_stdio(false); cin.tie(0);
20582
20583     int t; cin >> t;
20584     while (t--) {
20585         int n; cin >> n;
20586         vector<int> a(n);
20587         for (int i = 0; i < n; ++i) {
20588             cin >> a[i];
20589         }
20590         vector<vector<int>> g(n);
20591         for (int i = 0; i < n-1; ++i) {
20592             int u, v; cin >> u >> v;
20593             g[--u].push_back(--v);
20594             g[v].push_back(u);
20595         }
20596         int timer = 0;
20597         vector<int> in(n), out(n);
20598         vector<array<int, 18>> anc(n);
20599         auto dfs = [&] (const auto &self, int u, int p) -> void {
20600             in[u] = timer++;
20601             anc[u][0] = p;
20602             for (int i = 1; i < 18; ++i) anc[u][i] = anc[anc[u][i-1]][i-1];
20603             for (int v : g[u]) {
20604                 if (v == p) continue;
20605                 self(self, v, u);
20606             }
20607             out[u] = timer;
20608         };
20609         auto isanc = [&] (int u, int v) {return in[u] <= in[v] and out[u] >= out[v];};
20610         // Is u an ancestor of v?
20611         auto getchild = [&] (int u, int v) { // Which child of v contains u?
20612             for (int i = 17; i >= 0; --i) if (!isanc(anc[u][i], v)) u = anc[u][i];
20613             return u;
20614         };
20615         dfs(dfs, 0, 0);
20616
20617         Node *segtree = new Node(0, n);
20618         for (int i = 0; i < n; ++i) {
20619             segtree -> set(in[i], a[i]);
20620         }
20621         int q; cin >> q;
20622         int root = 0;
20623         while (q--) {
20624             int type; cin >> type;
20625             if (type == 1) {
20626                 int u, col; cin >> u >> col; --u;
20627                 if (u == root) segtree -> rangeset(0, n, col, true);
20628                 else if (in[u] <= in[root] and out[u] >= out[root]) {
20629                     int v = getchild(root, u);
20630                     segtree -> rangeset(out[v], n, col, true);
20631                     segtree -> rangeset(0, in[v], col, true);
20632                 }
20633                 else segtree -> rangeset(in[u], out[u], col, true);
20634             } else if (type == 2) {
20635                 cin >> root; --root;
20636             } else {
20637                 int u, col; cin >> u >> col; --u;
20638                 int ans = 0;
20639                 if (u == root) ans = segtree -> query(0, n, col);
20640                 else if (in[u] <= in[root] and out[u] >= out[root]) {

```

```

20640         int v = getchild(root, u);
20641         ans = segtree -> query(out[v], n, col) + segtree -> query(0, in[v],
20642                                     col);
20643     }
20644     else ans = segtree -> query(in[u], out[u], col);
20645     cout << ans << '\n';
20646 }
20647 }
20648 }
20649
20650 //EXISTENCEOFX
20651 #include <bits/stdc++.h>
20652 #include <ext/pb_ds/tree_policy.hpp>
20653 #include <ext/pb_ds/assoc_container.hpp>
20654 using namespace __gnu_pbds;
20655 using namespace std;
20656 #define ll long long
20657 const ll INF_MUL=1e13;
20658 const ll INF_ADD=1e18;
20659 #define pb push_back
20660 #define mp make_pair
20661 #define nline "\n"
20662 #define f first
20663 #define s second
20664 #define pll pair<ll,ll>
20665 #define all(x) x.begin(),x.end()
20666 #define vl vector<ll>
20667 #define vvl vector<vector<ll>>
20668 #define vvvl vector<vector<vector<ll>>>
20669 #ifndef ONLINE_JUDGE
20670 #define debug(x) cerr<<#x<<" "; _print(x); cerr<<nline;
20671 #else
20672 #define debug(x);
20673 #endif
20674 void _print(ll x){cerr<<x;}
20675 void _print(char x){cerr<<x;}
20676 void _print(string x){cerr<<x;}
20677 mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());
20678 template<class T,class V> void _print(pair<T,V> p) {cerr<<"{"; _print(p.first);cerr<<" ";
20679 ; _print(p.second);cerr<<"}";}
20680 template<class T>void _print(vector<T> v) {cerr<<" [ "; for (T i:v){_print(i);cerr<<" ";
20681 ;}cerr<<"]";}
20682 template<class T>void _print(set<T> v) {cerr<<" [ "; for (T i:v){_print(i); cerr<<" ";
20683 cerr<<"]";}
20684 template<class T>void _print(multiset<T> v) {cerr<<" [ "; for (T i:v){_print(i);cerr<<
20685 " ";}cerr<<"]";}
20686 template<class T,class V>void _print(map<T, V> v) {cerr<<" [ "; for(auto i:v) {_print(i
20687 );cerr<<" ";} cerr<<"]";}
20688 typedef tree<ll, null_type, less<ll>, rb_tree_tag, tree_order_statistics_node_update>
20689 ordered_set;
20690 typedef tree<ll, null_type, less_equal<ll>, rb_tree_tag,
20691 tree_order_statistics_node_update> ordered_multiset;
20692 typedef tree<pair<ll,ll>, null_type, less<pair<ll,ll>>, rb_tree_tag,
20693 tree_order_statistics_node_update> ordered_pset;
20694 //-----
20695 -----
20696 const ll MOD=998244353;
20697 const ll MAX=5000500;
20698 ll getv(ll x,ll bit){
20699     return min(1LL,x&(1LL<<bit));
20700 }
20701 void solve(){
20702     ll a,b,c,x,check; cin>>a>>b>>c;
20703     vector<ll> l(31,0),r(31,0);
20704     for(ll i=0;i<30;i++){
20705         ll checka=getv(a,i),checkb=getv(b,i),checkc=getv(c,i);
20706         if(checka!=checkb){

```

```

20698         l[i]++;
20699         if(l[i]%2){
20700             ;
20701         }
20702         else{
20703             l[i+1]++;
20704             l[i]=0;
20705         }
20706         r[i]=l[i];
20707     }
20708     else if(l[i]==checkc){
20709         r[i]=checkc;
20710         if(checka){
20711             l[i+1]++;
20712         }
20713     }
20714     else{
20715         r[i]=checkc^1;
20716         if(checka){
20717             ;
20718         }
20719         else{
20720             l[i+1]++;
20721         }
20722     }
20723 }
20724 if(l==r){
20725     cout<<"YES\n";
20726 }
20727 else{
20728     cout<<"NO\n";
20729 }
20730 return;
20731 }
20732 int main()

20733 {
20734     ios_base::sync_with_stdio(false);
20735     cin.tie(NULL);
20736     #ifndef ONLINE_JUDGE
20737     freopen("input.txt", "r", stdin);
20738     freopen("output.txt", "w", stdout);
20739     freopen("error.txt", "w", stderr);
20740     #endif
20741     ll test_cases=1;
20742     cin>>test_cases;
20743     while(test_cases--){
20744         solve();
20745     }
20746     cout<<fixed<<setprecision(10);
20747     cerr<<"Time:"<<1000*((double)clock())/((double)CLOCKS_PER_SEC<<"ms\n";
20748 }
20749
20750 //MUSROD
20751 #include <bits/stdc++.h>
20752 #include <iostream>
20753 #define ull long long int
20754 #define ll long long int
20755 using namespace std;
20756 #define maxlen 100
20757
20758 bool sortVec(const vector<ull> l, const vector<ull> r){
20759     if (l[1]*r[0] == l[0]*r[1]) return l[0] < r[0];
20760     else if (l[1]*r[0] > l[0]*r[1])return false;
20761     else return true;
20762 }
20763
20764

```

```

20765 void solve (ll k){
20766     ull n;
20767     cin>>n;
20768     vector<vector<ull>> rods(n, vector<ull>(2,0));
20769     ull sum = 0;
20770     for (ull i = 0; i < n; i++){
20771         cin>>rods[i][0];
20772         sum+=rods[i][0];
20773     }
20774     for (ull i = 0; i < n; i++){
20775         cin>>rods[i][1];
20776     }
20777
20778     //cout<<"here"<<"\n";
20779     sort(rods.begin(), rods.end(), sortVec);
20780     //cout<<"here"<<"\n";
20781     ull pos = 0;
20782     ull ans = 0;
20783     for (ull i = 0; i < n; i++){
20784         //cout<<pos<<" "<<rods[i][1]<<" "<<ans<<"\n";
20785         ans += pos*rods[i][1];
20786         pos += rods[i][0];
20787     }
20788     cout<<ans<<"\n";
20789 }
20790
20791 int main(){
20792     //ios_base::sync_with_stdio(false);
20793     //cin.tie(NULL);
20794     int t;
20795     cin>>t;
20796     for(int k=1;k<=t;k++){
20797         solve(k);
20798     }
20799     return 0;
20800 }
20801
20802 //INTERACTREE
20803 #pragma GCC optimize("O3")
20804 #pragma GCC target("popcnt")
20805 #pragma GCC target("avx,avx2,fma")
20806 #pragma GCC optimize("Ofast,unroll-loops")
20807 #include <bits/stdc++.h>
20808 #include <ext/pb_ds/tree_policy.hpp>
20809 #include <ext/pb_ds/assoc_container.hpp>
20810 using namespace __gnu_pbds;
20811 using namespace std;
20812 #define ll long long
20813 const ll INF_MUL=1e13;
20814 const ll INF_ADD=1e18;
20815 #define pb push_back
20816 #define mp make_pair
20817 #define nline "\n"
20818 #define f first
20819 #define s second
20820 #define pll pair<ll,ll>
20821 #define all(x) x.begin(),x.end()
20822 #define vl vector<ll>
20823 #define vvl vector<vector<ll>>
20824 #define vvvl vector<vector<vector<ll>>>
20825 #ifndef ONLINE_JUDGE
20826 #define debug(x) cerr<<#x<<" "; _print(x); cerr<<nline;
20827 #else
20828 #define debug(x);
20829 #endif
20830 void _print(ll x){cerr<<x;}
20831 void _print(int x){cerr<<x;}
20832 void _print(char x){cerr<<x;}

```

```

20834 void _print(string x){cerr<<x;}
20835 mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());
20836 template<class T,class V> void _print(pair<T,V> p) {cerr<<"{"; _print(p.first);cerr<<"",
; _print(p.second);cerr<<"}";}
20837 template<class T>void _print(vector<T> v) {cerr<<" [ "; for (T i:v){_print(i);cerr<<" "
;};cerr<<""]";}
20838 template<class T>void _print(set<T> v) {cerr<<" [ "; for (T i:v){_print(i); cerr<<" ";}
cerr<<""]";}
20839 template<class T>void _print(multiset<T> v) {cerr<<" [ "; for (T i:v){_print(i);cerr<<
" ";};cerr<<""]";}
20840 template<class T,class V>void _print(map<T, V> v) {cerr<<" [ "; for(auto i:v) {_print(i
);cerr<<" ";} cerr<<""]";}
20841 typedef tree<ll, null_type, less<ll>, rb_tree_tag, tree_order_statistics_node_update>
ordered_set;
20842 typedef tree<ll, null_type, less_equal<ll>, rb_tree_tag,
tree_order_statistics_node_update> ordered_multiset;
20843 typedef tree<pair<ll,ll>, null_type, less<pair<ll,ll>>, rb_tree_tag,
tree_order_statistics_node_update> ordered_pset;
20844 //-----
-----
20845 const ll MOD=998244353;
20846 const ll MAX=100100;
20847 ll query(vector<ll> v){
20848     cout<<"? "<<v.size()<<" ";
20849     for(auto it:v){
20850         cout<<it<<" ";
20851     }
20852     cout<<endl;
20853     ll q; cin>>q;
20854     return q;
20855 }
20856 void guess(ll x){
20857     cout<<"! "<<x<<endl;
20858 }
20859 vector<vector<pair<ll,ll>>> adj;
20860 ll dist[MAX],updist[MAX];
20861 void dfs(ll cur,ll par){
20862
20863     dist[cur] = updist[cur] = 0;
20864     for(auto it:adj[cur]){
20865         if(it.f!=par){
20866             dfs(it.f,cur);
20867             dist[cur]=max(dist[cur],dist[it.f]+1);
20868         }
20869     }
20870 }
20871 void dfs2(ll cur,ll par){
20872     multiset<ll> track; track.insert(-1);
20873     for(auto it:adj[cur]){
20874         if(it.f!=par){
20875             track.insert(dist[it.f]);
20876         }
20877     }
20878     for(auto it:adj[cur]){
20879         if(it.f!=par){
20880             track.erase(track.find(dist[it.f]));
20881             ll x=max(*(--track.end())+1,updist[cur]);
20882             updist[it.f]=x+1;
20883             dfs2(it.f,cur);
20884             track.insert(dist[it.f]);
20885         }
20886     }
20887 }
20888 void solve(){
20889     ll n; cin>>n;
20890     adj.clear(); adj.resize(n+5);
20891     for(ll i=1;i<n;i++){
20892         ll u,v; cin>>u>>v;

```

```

20893         adj[u].push_back({v,i});
20894         adj[v].push_back({u,i});
20895     }
20896     dist[1]=updist[1]=0;
20897     dfs(1,-1);
20898     dfs2(1,-1);
20899     vector<ll> v;
20900     ll q=query(v);
20901     vector<ll> track;
20902     for(ll i=1;i<=n;i++){
20903         if(max(dist[i],updist[i])==q){
20904             track.push_back(i);
20905         }
20906     }
20907     while(track.size()>1){
20908         ll len=track.size();
20909         ll mid=len/2;
20910         vector<ll> skip(n+5,0);
20911         for(ll i=0;i<mid;i++){
20912             for(auto it:adj[track[i]]){
20913                 skip[it.s]=1;
20914             }
20915         }
20916         v.clear();
20917         for(ll i=1;i<n;i++){
20918             if(skip[i]){
20919                 v.push_back(i);
20920             }
20921         }
20922         ll q=query(v);
20923         if(q==0){
20924             v.clear();
20925             for(ll i=0;i<mid;i++){
20926                 v.push_back(track[i]);
20927             }
20928             track=v;
20929         }
20930         else{
20931             v.clear();
20932             for(ll i=mid;i<len;i++){
20933                 v.push_back(track[i]);
20934             }
20935             track=v;
20936         }
20937     }
20938     guess(track[0]);
20939     return;
20940 }
20941 int main()
20942 {
20943     ios_base::sync_with_stdio(false);
20944     cin.tie(NULL);
20945     ll test_cases=1;
20946     cin>>test_cases;
20947     while(test_cases--){
20948         solve();
20949     }
20950     cout<<fixed<<setprecision(10);
20951     cerr<<"Time:"<<1000*((double)clock())/(double)CLOCKS_PER_SEC<<"ms\n";
20952 }
20953
20954 //INTERACTREE-EDITOR
20955 #include "bits/stdc++.h"
20956 // #pragma GCC optimize("O3,unroll-loops")
20957 // #pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
20958 using namespace std;
20959 using ll = long long int;
20960 mt19937_64 rng(chrono::high_resolution_clock::now().time_since_epoch().count());
20961

```

```

20962 int main()
20963 {
20964     ios::sync_with_stdio(false); cin.tie(0);
20965
20966     auto ask = [&] (auto ids) {
20967         cout << "? " << ids.size() << ' ';
20968         for (int x : ids) cout << x << ' ';
20969         cout << endl;
20970         int dist; cin >> dist;
20971         return dist;
20972     };
20973     auto ans = [&] (int x) {
20974         cout << "! " << x << endl;
20975     };
20976
20977     int t; cin >> t;
20978     while (t--) {
20979         int n; cin >> n;
20980         vector<vector<array<int, 2>>> adj(n+1);
20981         for (int i = 1; i < n; ++i) {
20982             int u, v; cin >> u >> v;
20983             adj[u].push_back({v, i});
20984             adj[v].push_back({u, i});
20985         }
20986         vector<int> paredge(n+1), dfsorder;
20987         auto dfs = [&] (const auto &self, int u, int p) -> void {
20988             dfsorder.push_back(u);
20989             for (auto [v, id] : adj[u]) {
20990                 if (v == p) continue;
20991                 paredge[v] = id;
20992                 self(self, v, u);
20993             }
20994         };
20995         dfs(dfs, 1, 0);
20996         int lo = 1, hi = n-1;
20997         while (lo < hi) {
20998             int mid = (lo + hi + 1)/2;
20999             // Isolate all vertices from mid onwards
21000             vector<int> ids;
21001             for (int i = mid; i < n; ++i) ids.push_back(paredge[dfsorder[i]]);
21002             if (ask(ids)) hi = mid-1;
21003             else lo = mid;
21004         }
21005         if (lo > 1) {
21006             ans(dfsorder[lo]);
21007             continue;
21008         }
21009         int u = dfsorder[0], v = dfsorder[1];
21010         if (adj[u].size() > adj[v].size()) swap(u, v);
21011         vector<int> ids;
21012         for (auto [x, id] : adj[u]) ids.push_back(id);
21013         if (ask(ids)) ans(v);
21014         else ans(u);
21015     }
21016 }
21017
21018 //MAXAGRY
21019 #include<iostream>
21020 #include<iterator>
21021 #include<algorithm>
21022 #include<bits/stdc++.h>
21023
21024 using namespace std;
21025
21026 typedef long long int ll;
21027 typedef long double ld;
21028 typedef std::vector<int> vi;
21029 typedef std::vector<ll> vll;
21030 typedef std::vector<ld> vld;

```

```

21031 typedef std::vector<std::vector<ll> > vvll;
21032 typedef std::vector<std::vector<ld> > vvld;
21033 typedef std::vector<std::vector<std::vector<ll> > > vvvll;
21034 typedef std::vector<string> vstr;
21035 typedef std::vector<std::pair<ll,ll> > vp11;
21036 typedef std::pair<ll,ll> p11;
21037
21038 #define f(i_itr,a,n) for(ll i_itr=a; i_itr<n; i_itr++)
21039 #define rev_f(i_itr,n,a) for(ll i_itr=n; i_itr>a; i_itr--)
21040
21041 #define pb push_back
21042 #define fi first
21043 #define se second
21044 #define all(a) a.begin(),a.end()
21045
21046 #define ms(a,val) memset(a,val,sizeof(a))
21047
21048 const ll mod = 1000000007;
21049 const ll N = 1e5 + 5;
21050
21051 ll setBitNumber(int n)
21052 {
21053     // calculate the number
21054     // of trailing zeroes
21055     ll k = __builtin_clz(n);
21056
21057     // To return the value
21058     // of the number with set
21059     // bit at (31 - k)-th position
21060     // assuming 32 bits are used
21061     return 1 << (31 - k);
21062 }
21063
21064 void solve()
21065 {
21066     ll n, k;
21067     cin >> n >> k;
21068     if (k >= n / 2) {
21069         cout << (n * (n - 1)) / 2<<endl;
21070         return;
21071     }
21072
21073     ll t = n - k * 2;
21074     cout << (n * (n - 1)) / 2 - (t * (t - 1)) / 2<<endl;
21075 }
21076
21077 int main()
21078 {
21079     ios_base::sync_with_stdio(false);
21080     cin.tie(NULL);
21081
21082     ll qq_itr=1;
21083     cin >> qq_itr;
21084     while (qq_itr--)
21085         solve();
21086     return 0;
21087 }
21088
21089 //TILL START67
21090 //TO START 66
21091
21092
21093

```