#include <bits/stdc++.h>

#define mp make\_pair

#define pb push\_back

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

#define all(x) begin(x), end(x)

#define fi first

#define se second

#define debug(x) cerr << #x << " " << x << '\n'

using namespace std;

using ll = long long;

using pii = pair<int,int>;

using pli = pair<ll,int>;

const int INF = 0x3f3f3f3f, N = 20;

const ll LINF = 1e18 + 5;

constexpr int mod = 1e9 + 7;

int n, m, p[N], pos[N], x[N], y[N];

int tmpp[N];

int f[N], ans;

vector<vector<int>> vec;

vector<int> rt[N];

int find(int x)

{

if(x==f[x]) return x;

return f[x] = find(f[x]);

}

void merge(int x, int y)

{

x = find(x), y = find(y);

if(x==y) return;

f[x] = y;

}

int cal()

{

int cnt = 0;

for(int i=1; i<=n; i++)

tmpp[i] = p[i];

for(int i=1; i<=n; i++)

{

int j = i;

while(pos[tmpp[j]]&&pos[tmpp[j]]!=j)

{

cnt++;

swap(tmpp[j], tmpp[pos[tmpp[j]]]);

}

}

return cnt;

}

void dfs(int x)

{

if(x==sz(vec))

{

ans = min(ans, cal());

return;

}

vector<int> idx(sz(vec[x]));

iota(all(idx), 0);

do {

for(int i=0; i<sz(vec[x]); i++)

pos[vec[x][i]] = vec[x][idx[i]];

dfs(x+1);

}while(next\_permutation(all(idx)));

}

void solve()

{

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

ans = n - 1;

vec.clear();

for(int i=1; i<=n; i++)

{

scanf("%d", p+i);

f[i] = i;

pos[i] = 0;

rt[i].clear();

}

for(int i=1; i<=m; i++)

{

scanf("%d%d", x+i, y+i);

merge(x[i], y[i]);

}

int mx = 0;

for(int i=1; i<=n; i++)

{

int r = find(i);

rt[r].pb(i);

mx = max(mx, sz(rt[r]));

}

bool tag = 0;

for(int i=1; i<=n; i++)

if(f[i]==i && sz(rt[i])<mx) vec.pb(rt[i]);

else if(f[i]==i && sz(rt[i])==mx)

{

if(!tag) tag = 1;

else vec.pb(rt[i]);

}

dfs(0);

printf("%d\n", ans);

}

int main()

{

int T;

scanf("%d", &T);

while(T--) solve();

return 0;

}