1. *#include* <iostream>
2. *#include* <algorithm>
3. *#include* <vector>
4. *#include* <set>
5. *#define* MAXN 100010
7. using namespace std;
9. int gcd(int a, int b) {
10. while (a && b) {
11. if (a > b) a %= b;
12. else b %= a;
13. }
14. return a + b;
15. }
17. int n,m;
18. int c[MAXN];
19. vector<int> e[MAXN];
20. int id[MAXN],lo[MAXN],instk[MAXN], curid;
21. vector<int> stk;
22. int scc[MAXN], numscc, cc[MAXN];
23. vector<int> ee[MAXN];
24. set<int> s[MAXN];
26. void dfs(int u) {
27. id[u] = lo[u] = ++curid;
28. instk[u] = 1;
29. stk.push\_back(u);
30. for (int v : e[u]) {
31. if (!id[v]) {
32. dfs(v);
33. lo[u] = min(lo[u], lo[v]);
34. } else if (instk[v]) {
35. lo[u] = min(lo[u], id[v]);
36. }
37. }
39. if (lo[u] == id[u]) {
40. numscc++;
41. while (1) {
42. int v = stk.back();
43. stk.pop\_back();
44. instk[v] = 0;
45. scc[v] = numscc;
46. cc[numscc] = gcd(cc[numscc], c[v]);
47. for (int w : e[v])
48. if (scc[w] && scc[w] != numscc)
49. ee[numscc].push\_back(scc[w]);
50. if (v == u) break;
51. }
52. }
53. }
55. void solve(int u) {
56. if (!s[u].empty()) return;
57. s[u].insert(cc[u]);
58. for (int v : ee[u]) {
59. solve(v);
60. for (int g : s[v]) {
61. s[u].insert(gcd(cc[u], g));
62. }
63. }
64. }
66. int main() {
67. ios::sync\_with\_stdio(0);
68. cin >> n >> m;
69. for (int i = 1; i <= n; i++) {
70. cin >> c[i];
71. }
72. for (int i = 0; i < m; i++) {
73. int u,v;
74. cin >> u >> v;
75. e[u].push\_back(v);
76. }
78. for (int i = 1; i <= n; i++) {
79. if (!id[i]) dfs(i);
80. }
82. int ans = MAXN;
83. for (int i = 1; i <= numscc; i++) {
84. solve(i);
85. ans = min(ans, \*s[i].begin());
86. }
88. cout << ans << '\n';
89. }

**Language:**C++

1. *#include* <bits/stdc++.h>
2. *#define* ALL(a) (a).begin(), (a).end()
3. *#define* SZ(a) ((int)(a).size())
4. *#define* FIN ios\_base::sync\_with\_stdio(0); cin.tie(0); cout.tie(0)
5. *#define* pb push\_back
6. *#define* fore(i,a,b) for(int i = a; i < b; i++)
7. using namespace std;
8. typedef long long ll;
10. const int MXN = 1e5;
11. vector<int> g[2][MXN];
12. vector<int> f[MXN];
13. int N, M, vis[MXN], scc\_cnt, C[MXN], CC[MXN];
14. stack<int> st; int super\_cnt = 0, mx\_super\_cnt = 5e8;
16. void dfs(int u, int m=-1){
17. vis[u] = max(m,0);
18. for(int v : g[min(m+1,1)][u])
19. if(vis[v]==-1)
20. dfs(v,m);
21. if(m==-1) st.push(u);
22. }
24. void gen\_scc(){
25. scc\_cnt = 0;
26. memset(vis,-1,sizeof(vis));
27. fore(x,0,N) if(vis[x]==-1) dfs(x);
28. memset(vis,-1,sizeof(vis));
29. while(!st.empty()){
30. int u=st.top(); st.pop();
31. if(vis[u]==-1)
32. dfs(u,scc\_cnt),scc\_cnt++;
33. }
34. }
36. map<int,int> dp[MXN];
38. int solve(int i, int g){
39. if(dp[i].count(g)) return dp[i][g];
40. int ans=g;
41. for(int v : f[i])
42. ans = min(ans, solve(v,\_\_gcd(g,CC[v])));
43. return dp[i][g] = ans;
44. }
46. int main(){FIN;
47. cin >> N >> M;
48. fore(x,0,N) cin >> C[x];
49. fore(x,0,M){
50. int a,b; cin >> a >> b;a--; b--;
51. g[0][a].pb(b); g[1][b].pb(a);
52. }
53. gen\_scc();
54. fore(u,0,N){
55. for(int v : g[0][u])
56. if(vis[v] != vis[u])
57. f[vis[u]].pb(vis[v]);
58. CC[vis[u]]=\_\_gcd(CC[vis[u]],C[u]);
59. }
60. fore(u,0,scc\_cnt){
61. sort(ALL(f[u]));
62. f[u].resize(distance(f[u].begin(),unique(ALL(f[u]))));
63. }
64. int ans=1e9;
65. fore(x,0,scc\_cnt)
66. ans=min(ans,solve(x,CC[x]));
67. cout << ans << "\n";
68. }

**Language:**C++