

D. Fair

time limit per test 2 seconds
 memory limit per test 512 megabytes
 input standard input
 output standard output

Some company is going to hold a fair in Byteland. There are n towns in Byteland and m two-way roads between towns. Of course, you can reach any town from any other town using roads.

There are k types of goods produced in Byteland and every town produces only one type. To hold a fair you have to bring at least s different types of goods. It costs $d(u, v)$ coins to bring goods from town u to town v where $d(u, v)$ is the length of the shortest path from u to v . Length of a path is the number of roads in this path.

The organizers will cover all travel expenses but they can choose the towns to bring goods from. Now they want to calculate minimum expenses to hold a fair in each of n towns.

Input

There are 4 integers n, m, k, s in the first line of input ($1 \leq n \leq 10^5, 0 \leq m \leq 10^5, 1 \leq s \leq k \leq \min(n, 100)$) — the number of towns, the number of roads, the number of different types of goods, the number of different types of goods necessary to hold a fair.

In the next line there are n integers a_1, a_2, \dots, a_n ($1 \leq a_i \leq k$), where a_i is the type of goods produced in the i -th town. It is guaranteed that all integers between 1 and k occur at least once among integers a_i .

In the next m lines roads are described. Each road is described by two integers $u \ v$ ($1 \leq u, v \leq n, u \neq v$) — the towns connected by this road. It is guaranteed that there is no more than one road between every two towns. It is guaranteed that you can go from any town to any other town via roads.

Output

Print n numbers, the i -th of them is the minimum number of coins you need to spend on travel expenses to hold a fair in town i . Separate numbers with spaces.

Examples

input	Copy
<pre>5 5 4 3 1 2 4 3 2 1 2 2 3 3 4 4 1 4 5</pre>	
output	Copy
<pre>2 2 2 2 3</pre>	

input	Copy
<pre>7 6 3 2 1 2 3 3 2 2 1 1 2 2 3 3 4 2 5 5 6 6 7</pre>	
output	Copy
<pre>1 1 1 2 2 1 1</pre>	

Note

Let's look at the first sample.

To hold a fair in town 1 you can bring goods from towns 1 (0 coins), 2 (1 coin) and 4 (1 coin). Total numbers of coins is 2.

Town 2: Goods from towns 2 (0), 1 (1), 3 (1). Sum equals 2.

Town 3: Goods from towns 3 (0), 2 (1), 4 (1). Sum equals 2.

Town 4: Goods from towns 4 (0), 1 (1), 5 (1). Sum equals 2.

Town 5: Goods from towns 5 (0), 4 (1), 3 (2). Sum equals 3.

```
1 // http://codeforces.com/contest/987/problem/D
2 #include<bits/stdc++.h>
3 using namespace std;
4 #define pii pair<int,int>
5 #define mp make_pair
6 #define MAXN 100005
7 #define INF 10000000
8 int n,m,s,k;
9 int ans[MAXN],dp[MAXN][105];
10 vector<int>ed[MAXN];
11 vector<int>vv[105];
12 void BFS(int c){
13     for(int i=1; i<=n; i++) dp[i][c] = INF;
14     queue<pii>Q;
15     for(int i=0;i<vv[c].size(); i++){
16         Q.push(mp(vv[c][i],0));
17         dp[vv[c][i]][c] = 0;
18     }
19
20     while(!Q.empty()){
21         pii nd = Q.front();
22         Q.pop();
23
24         int u = nd.first;
25         int w = nd.second;
26
27         for(int i=0; i<ed[u].size(); i++){
28             int v = ed[u][i];
29             if(dp[v][c]>dp[u][c]+1){
30                 dp[v][c] = dp[u][c]+1;
31                 Q.push(mp(v,dp[v][c]));
32             }
33         }
34     }
35 }
36
37 int main(){
38     ios::sync_with_stdio(false); cin.tie(0);
39     cin>>n>>m>>k>>s;
40     for(int i=1; i<=n; i++){
41         int x; cin>>x;
42         vv[x].push_back(i);
43     }
44
45     for(int i=1; i<=m; i++){
46         int u,v; cin>>u>>v;
47         ed[u].push_back(v);
48         ed[v].push_back(u);
49     }
50
51     for(int i=1; i<=k; i++) BFS(i);
52
53     for(int i=1; i<=n; i++) sort(dp[i]+1,dp[i]+k+1);
54
55     for(int i=1; i<=n; i++){
56         ans[i] = 0;
57         for(int j=1; j<=s; j++){
58             ans[i]+=dp[i][j];
59         }
60     }
61
62     for(int i=1; i<=n; i++){
63         if(i<n)cout << ans[i] << " ";
64         else cout << ans[i] << endl;
65     }
66     return 0;
67 }
```