Talent Battle 100 Days Coding Series

You are given an undirected graph with *N* nodes (numbered 1 through *N*) and *M* edges. Each edge connects two distinct nodes. However, there may be multiple edges connecting the same pairs of nodes, and they are considered to be distinct edges. A lowercase English letter is written in each node.

You are also given a string *S* with length *L*. A *beautiful path* is a sequence of *L*–1 edges such that there is a sequence of *L* nodes with the following properties:

- for each valid *i*, the *i*-th edge connects the *i*-th and (*i*+1)-th of these nodes
- for each valid *i*, the *i*-th character of *S* is written in the *i*-th of these nodes

There are no other restrictions — a path may visit nodes or edges any number of times in any order.

Determine the number of beautiful paths in the graph. Since the answer can be very large, compute it modulo (10^9)+7.

Input

- The first line of the input contains a single integer *T* denoting the number of test cases. The description of *T* test cases follows.
- The first line of each test case contains three space-separated integers *N*, *M* and *L*.
- The second line contains a single string *S* with length *L*.
- The third line contains a single string with length *N*. For each valid *i*, the *i*-th character of the string is the letter written in the *i*-th node.
- Two lines follow. The first line contains *M* integers *u*1,…,*um*. The second lines contains *M* integers, *v*1,…,*vm*. This denotes that there is an edge connecting nodes *ui* and to *vi*. These edges are distinct, even though they may connect the same pair of nodes.

Output

For each test case, print a single line containing one integer — the number of beautiful paths modulo (10^9)+7.

Sample Input

2

4 4 3

aac

aaca

1 2 2 1

2 3 4 2

2 1 2

aa

aa

1

2

Sample Output

3

1

C++

```
#include <bits/stdc++.h>
using namespace std;
#define int long long
void read (int &x) {
   char ch = getchar(); x = 0; while (!isdigit(ch)) ch = getchar();
   while (isdigit(ch)) x = x * 10 + ch - 48, ch = getchar();
} const int N = 1024, mod = 1e9 + 7;
int n, m, l, f[22][N], x[N * 2], y[N * 2]; char a[N], b[N];
int cnt[N][N];
signed main() {
   int T; read (T);
   while (T--) {
      memset (f, 0, sizeof (f));
      read (n), read (m), read (l);
      scanf ("%s %s", b + 1, a + 1);
      for (int i = 1; i <= m; ++i) read (x[i]);
      for (int i = 1; i <= m; ++i) read (y[i]);
      for (int i = 1; i <= m; ++i) if (x[i] > y[i]) swap (x[i], y[i]);
      for (int i = 1; i <= m; ++i) x[i + m] = y[i], y[i + m] = x[i];
```

```
    for (int i = 1; i <= n; ++i) if (a[i] == b[1]) f[1][i] = 1;
    for (int i = 2; i <= l; ++i) {
      for (int j = 1; j <= m * 2; ++j)
        if (a[y[j]] == b[i]) (f[i][y[j]] += f[i - 1][x[j]]) %= mod;
    }
    int res = 0;
    for (int i = 1; i <= n; ++i) (res += f[l][i]) %= mod;
    for (int i = 1; i <= m; ++i) ++cnt[x[i]][y[i]];
    int tag = 1;
    for (int i = 2; i <= l; ++i) if (b[i] != b[1]) tag = 0;
    if (tag) {
      for (int i = 1; i <= m; ++i) {
        if (cnt[x[i]][y[i]] && a[x[i]] == b[1] && a[y[i]] == b[1]) {
          int ans = 1;
          for (int j = 1; j < l; ++j) (ans *= cnt[x[i]][y[i]]) %= mod;
          (res -= ans) %= mod;
        }
        cnt[x[i]][y[i]] = 0;
      }
    }
    for (int i = 1; i <= m; ++i) cnt[x[i]][y[i]] = 0;
    printf("%lld\n", (res + mod) % mod);
  }
  return 0;
}
```

Java

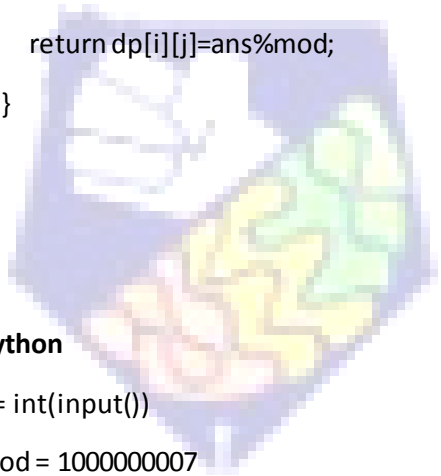
```
import java.util.*;
import java.lang.*;
import java.io.*;


class Main
{    static ArrayList<Integer> tree[];
  static int f[][];
   public static void main(String[] args) {
      Scanner input=new Scanner(System.in);
      int t=input.nextInt();
      while (t-->0){
        int n=input.nextInt();
        int m=input.nextInt();
        int l=input.nextInt();
        String s=input.next();
        char a[]=input.next().toCharArray();
        tree=new ArrayList[n+1];
        for (int i = 0; i <=n ; i++) {
           tree[i]=new ArrayList<>();
        }
        int x[]=new int[m];
        for (int i = 0; i <m ; i++) {
           x[i]=input.nextInt();
        }
        f=new int[n+1][n+1];
        for (int i = 0; i <m ; i++) {
           int y=input.nextInt();
```

```
    tree[x[i]].add(y);
    tree[y].add(x[i]);
    f[x[i]][y]++;
    f[y][x[i]]++;
}

long res=0;
dp=new Long[n+2][22];

for (int i = 1; i <=n ; i++) {
   res+= dfs(i,0,s,l,a);
   res%=mod;
}
boolean allsame=true;
for (int i = 1; i <l ; i++) {
    if (s.charAt(i) !=s.charAt(i-1)) allsame=false;
}
if (allsame){
    long temp=0;
    dp2=new Long[n+1][n+1];
    boolean v[][]=new boolean[n+1][n+1];
    for (int i = 1; i <=n ; i++) {
        for (int c:tree[i]) {
            if (v[i][c]) continue;
            if (a[i-1]==a[c-1]) {
                v[i][c]=true;
                v[c][i]=true;
                temp+=power(f[i][c],l-1,mod);
                temp%=mod;
```

```
            }
          }
        }


        System.out.println((res-temp+mod)%mod);
      }else {
        System.out.println(res);
      }

    }
}
static  Long dp[][];
static  Long dp2[][];
static long mod= (long) (1e9+7);
private static long dfs2(int i, int j, String s, int l, char a[],int k) {
    if (j==l-1){
        if (s.charAt(j)!=a[i-1]) return 0;
        return 1;
    }
    if (s.charAt(j)!=a[i-1]) return 0;
    if (dp2[i][j]!=null) return dp2[i][j];

    long ans=0;
    for (int c:tree[i]) {
        if (c!=k) continue;

        ans+=dfs2(k, j+1, s, l, a,i)%mod;
```

```
        ans%=mod;

    }


    return dp2[i][j]=ans%mod;


}
static long power(long x,
                  long y, long p)
{
    long res = 1;
    x = x % p;

    while (y > 0)
    {
        if ((y & 1) > 0)
            res = (res * x) % p;
        y = y >> 1;
        x = (x * x) % p;
    }
    return res;
}


private static long dfs(int i, int j, String s, int l, char[] a) {
    if (j==l-1){
        if (s.charAt(j)!=a[i-1]) return 0;
```

```
            return 1;
        }
        if (s.charAt(j)!=a[i-1]) return 0;
        if (dp[i][j]!=null) return dp[i][j];

        long ans=0;
        for (int c:tree[i]) {
            ans+=((dfs(c, j+1, s, l, a)))%mod;
            ans%=mod;

        }
        return dp[i][j]=ans%mod;
    }

}
```



Python

```
t = int(input())
mod = 1000000007
for _ in range(t):
    n, m, l = list(map(int, input().split()))
    s = input()
    v = input()
    e1 = list(map(int, input().split()))
    e2 = list(map(int, input().split()))
    adj = [[] for _ in range(n + 1)]
    count = {}
    for i in range(m):
        adj[e1[i] - 1].append(e2[i] - 1)
```

```
    adj[e2[i] - 1].append(e1[i] - 1)
    temp = [e1[i]-1, e2[i]-1]
    temp.sort()

    if tuple(temp) not in count:
        count[tuple(temp)] = 0
    count[tuple(temp)] += 1

dp = [[0 for _ in range(l)] for _ in range(n)]
for j in range(0, n):
    if s[0] == v[j]:
        dp[j][0] = 1

for j in range(1, l):
    for i in range(0, n):
        if s[j] != v[i]:
            continue
        for k in adj[i]:
            dp[i][j] = (dp[i][j] + dp[k][j - 1]) % mod
ans = 0
for i in range(0, n):
    ans = (ans + dp[i][l - 1]) % mod

if min(s) == max(s):
    for i in range(0, n):
        for j in range(i + 1, n):
            if v[i] == v[j] and (i,j) in count:
                ans = (ans - pow(count[(i, j)], l - 1, mod) + mod) % mod
print(ans)
```