[Home](https://www.codechef.com/) » [Compete](https://www.codechef.com/contests/) » [August Cook-Off 2020 Division 1](https://www.codechef.com/COOK121A?order=desc&sortBy=successful_submissions) » Red-blue Trees

**Red-blue Trees**Problem Code: **RBTREES**

******

[Submit (Practice)](https://www.codechef.com/submit/RBTREES)

[Tweet](https://twitter.com/share)

**Read problems statements in**[**Hindi**](http://www.codechef.com/download/translated/COOK121/hindi/RBTREES.pdf)**,**[**Mandarin Chinese**](http://www.codechef.com/download/translated/COOK121/mandarin/RBTREES.pdf)**,**[**Russian**](http://www.codechef.com/download/translated/COOK121/russian/RBTREES.pdf)**,**[**Vietnamese**](http://www.codechef.com/download/translated/COOK121/vietnamese/RBTREES.pdf)**, and**[**Bengali**](http://www.codechef.com/download/translated/COOK121/bengali/RBTREES.pdf)**as well.**

In a *red-blue tree*, each vertex is either red or blue and adjacent vertices always have different colours.

You are given a tree with NN vertices (numbered 11 through NN). It is not necessarily a red-blue tree, but its vertices are still coloured red and blue. You may perform the following operation any number of times (including zero): choose two adjacent vertices and swap their colours.

Find the smallest number of operations required to transform the tree into a red-blue tree or determine that it is impossible.

**Input**

* The first line of the input contains a single integer TT denoting the number of test cases. The description of TT test cases follows.
* The first line of each test case contains a single integer NN.
* N−1N−1 lines follow. Each of these lines contains two space-separated integers uu and vv denoting that vertices uu and vv are connected by an edge.
* The last line contains a string SS with length NN. For each valid ii, the ii-th character of this string is either '0' if the ii-th vertex is initially red or '1' if it is initially blue.

**Output**

Print a single line containing one integer ― the smallest number of operations or −1−1 if it is impossible to transform the tree into a red-blue tree.

**Constraints**

* 1≤T≤1001≤T≤100
* 1≤N≤100,0001≤N≤100,000
* 1≤u,v≤N1≤u,v≤N
* SS contains only characters '0' and '1'
* the sum of NN over all test cases does not exceed 106106

**Example Input**

1

7

1 2

1 3

2 4

2 5

3 6

3 7

0010010

**Example Output**

3

**Explanation**

**Example case 1:** We can perform the following operations:

* Swap the colours of vertices 11 and 33; the string of colours becomes "1000010".
* Swap the colours of vertices 11 and 22; the string of colours becomes "0100010".
* Swap the colours of vertices 66 and 33; the string of colours becomes "0110000", so the tree is a red-blue tree.

All submissions for this problem are available.

Author:[5★pandey\_\_ji](https://www.codechef.com/users/pandey__ji)

Editorial:<https://discuss.codechef.com/problems/RBTREES>

Tags:[bipartiteness](https://www.codechef.com/tags/problems/bipartiteness), [cook121](https://www.codechef.com/tags/problems/cook121), [dfs](https://www.codechef.com/tags/problems/dfs), [easy-medium](https://www.codechef.com/tags/problems/easy-medium), [greedy](https://www.codechef.com/tags/problems/greedy), [pandey\_\_ji](https://www.codechef.com/tags/problems/pandey__ji), [pandey\_\_ji](https://www.codechef.com/tags/problems/pandey__ji), [psychik](https://www.codechef.com/tags/problems/psychik)

Date Added:15-08-2020

Time Limit:1.5 secs

Source Limit:50000 Bytes

Languages:CPP14, C, JAVA, PYTH 3.6, PYTH, CS2, ADA, PYPY, PYP3, TEXT, CPP17, PAS fpc, RUBY, PHP, NODEJS, GO, TCL, HASK, PERL, SCALA, kotlin, BASH, JS, PAS gpc, BF, LISP sbcl, CLOJ, LUA, D, R, CAML, rust, ASM, FORT, FS, LISP clisp, SQL, swift, SCM guile, PERL6, CLPS, WSPC, ERL, ICK, NICE, PRLG, ICON, PIKE, COB, SCM chicken, SCM qobi, ST, NEM, SQLQ

#include<bits/stdc++.h>

#define pb push\_back

#define pii pair<int,int>

#define int long long int

#define vec vector<int>

#define mp map<int,int>

#define inf 1e18

using namespace std;

vector<int> adj[100001];

int parity[100001],red[100001],black[100001],color[100001];

void assignparity(int node,int par,int p)

{

parity[node]=p;

for(int child: adj[node])

{

if(child==par)

continue;

assignparity(child,node,p^1);

}

}

int ans;

void dfs(int node,int par,int p)

{

black[node]=0,red[node]=0;

if(parity[node]==0 && color[node]!=p)

black[node]=1;

if(parity[node]==1 && color[node]!=p^1)

red[node]=1;

for(int child: adj[node])

{

if(child==par)

continue;

dfs(child,node,p);

black[node]+=black[child];

red[node]+=red[child];

}

int temp=min(black[node],red[node]);

black[node]-=temp;

red[node]-=temp;

ans+=black[node]+red[node];

}

int32\_t main()

{

ios\_base::sync\_with\_stdio(false);

cin.tie(NULL);

cout.tie(NULL);

int tt=1;

cin>>tt;

while(tt--)

{

for(int i=0;i<100001;i++)

adj[i].clear(),red[i]=0,parity[i]=0,black[i]=0,color[i]=0;

int n;

cin>>n;

for(int i=0;i<n-1;i++)

{

int u,v;

cin>>u>>v;

adj[u].pb(v);

adj[v].pb(u);

}

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

{

char c;

cin>>c;

color[i]=c-'0';

}

assignparity(1,-1,0);

int res=inf;

ans=0;

dfs(1,-1,0); // assigning red color(0) to parity 0 elements

if(red[1]==black[1])

res=min(ans,res);

ans=0;

dfs(1,-1,1); // assigning red color(0) to parity 1 elements

if(red[1]==black[1])

res=min(ans,res);

if(res!=inf)

cout<<res<<"\n";

else

cout<<"-1\n";

}

}

/\*

3

7

1 2

1 3

2 4

2 5

3 6

3 7

1101101

10

7 8

8 6

6 10

10 2

2 3

2 9

9 4

9 1

7 5

0011100010

13

7 6

6 13

13 8

7 3

3 9

9 11

11 4

6 1

1 2

2 10

13 12

12 5

1001101001100

\*/