

T1

很容易证明答案为 $(x_2 - x_1) * (y_2 - y_1) + 1$ 。每一个值都可以取到

T2

区间dp

$dp[l][r]$ 表示在删除字符串 $l \sim r$ 区间所需的最小次数。

考虑左边的字符要么自己删除，要么跟区间中相同的字符一起删除进行转移

```

int dp(int i,int j){
    int & ans = d[i][j];
    if(ans > 0 )return ans;
    if(i > j) return 0;
    if(i == j) return ans = 1;
    ans = min(dp(i+1,j),dp(i,j-1)) + 1;//两边往里
    删
    for(int k = i+1;k<=j;k++)
        if(a[i] == a[k] ) ans = min(ans,dp(i+1,k-
1)+dp(k,j));
    //先删除i+1->k-1, 然后k->j也是一个子结构, 无论怎
    么删, i都可以跟k一起删除
    return ans;
}

```

T3

$dp[i]$ 表示i子树最少有几个叶子可以影响自己

如果根是max, 那么肯定选择子树中叶子节点最少的

如果根是min, 那么全都会影响自己

最终答案是 $cnt - dp[1] + 1$

```

#include<bits/stdc++.h>
using namespace std;
using ll=long long;
vector<int>v[300010];
int k[300030];
int a[300030];
int cnt=0;
int ans=0;
int temp=0;

void dfs(int x){
    for (int i:v[x]){
        dfs(i);
        if (k[x])a[x]=min(a[i],a[x]);
        else a[x]+=a[i];
    }
}

int main(){
    int n;cin>>n;
    for (int i=1;i<=n;i++)cin>>k[i];
    for (int i=2;i<=n;i++){
        int x;cin>>x;
        v[x].push_back(i);
    }
    for (int i=1;i<=n;i++)
        if (v[i].empty()){cnt++;a[i]=1;}
        else if (k[i])a[i]=1e9;
}

```

```
dfs(1);  
cout<<cnt-a[1]+1<<endl;  
}
```