N individuals are kidnapped and thrown in an arena to play a game. In this game, they must face a fearsome monster that has remained undefeated for thousands of years. They are required to approach the monster sequentially with the leftmost person being the first one. Each of the individuals possesses a unique vulnerability level A_i where ($1 <= A_i <= N$). They can only defeat the monster if they approach the monster in an ascending order of their vulnerability levels.

They are allowed to perform a certain kind of swap operation which may help them achieve victory. They are given a string S of N-1 characters where S_i = 1 denotes that the person at the i-th position can swap their position with a person at the (i + 1)-th position and S_i = 0 denotes that the person at the i-th position cannot swap their position with the (i + 1)-th one.

You are required to find out whether this group of individuals will be able to defeat the monster using any sequence (the sequence can be empty as well) of operations or will the monster still remain undefeated.

Input

- First line contains an integer N ($2 <= N <= 100000$) - number of individuals.
- Second line contains N unique integers $A_1, A_2, \ldots, A_N$ ($1 <= A_i <= N$) - vulnerability level of the individuals.
- The third line contains a string S of N - 1 characters, each character is either 0 or 1. If i-th character is 1, then you can swap the i-th player with the (i + 1)-th player any number of times, otherwise it is not possible to swap the i-th player with the (i + 1)-th one.

Output

If it is possible to defeat the monster using any sequence of swaps, print ‘YES’ (without the quotes). Otherwise print ‘NO’ (without the quotes).

Sample Input

```
6
1 2 5 3 4 6
01110
```

Sample Output

```
YES
```

C++

```cpp
#include <bits/stdc++.h>
using namespace std;

int main() {
    int n;
    cin >> n;
    vector<int> a(n);
    for (int i = 0; i < n; i++) {
        cin >> a[i];
    }
    vector<char> s(n);
    for (int i = 0; i < (n - 1); i++) {
        cin >> s[i];
    }

    vector<int> tmp;
    vector<vector<int>> v;
    for (int i = 0; i < n; i++) {
        tmp.push_back(a[i]);
        if (s[i] == '0') {
            sort(tmp.begin(), tmp.end());
            v.push_back(tmp);
            tmp.clear();
        }
    }

    vector<int> b;
    for (int i = 0; i < v.size(); i++) {
        for (int j = 0; j < v[i].size(); j++) {
            b.push_back(v[i][j]);
        }
    }

    if (is_sorted(b.begin(), b.end())) {
        cout << "YES" << endl;
    }
    else {
        cout << "NO" << endl;
    }
}
```

Python

```python
n = int(input())
i = 0
a = [None]*n
for num in list(map(int, input().split())):
    a[i] = num
    i += 1
i = 0
s = [None]*n
for num in list(input()):
    s[i] = num
    i += 1

tmp = []
v = []
for i in range(n):
    tmp.append(a[i])
    if s[i] == '0':
        tmp.sort()
        v.append(tmp)
        tmp = []

b = []
for i in range(len(v)):
    for j in range(len(v[i])):
        b.append(v[i][j])

if b == sorted(b):
    print("YES")
else:
    print("NO")
```