COMPUTER SOCIETY OF INDIA (CSI)
COEP TECH STUDENT CHAPTER

PRESENTS

CODE QUEST 5.0

EDITIORIAL

GOKU'S ALGORYTHMIA EXPEDITION

PROBLEM STATEMENT

Goku is a professional hiker and has completed many hiking expeditions around the world. He also makes vlogs when he travels and shares his experience about the hikes. He is planning to visit the city of Algorythmia which is surrounded by many Mighty Forts.

There are N forts surrounding the beautiful city of Algorythmia (labelled from 1 to N), connected by N-1 bidirectional roads (labelled from 1 to N-1). The i-th road connects Fort Fi to Fort Gi. Each road connects exactly two forts, and no two roads connect the same pair of forts. The Engineers of Algorythmia have constructed these roads in minimalistic way such that there is only a single sequence of roads connecting any two forts surrounding Algorythmia.

PROBLEM STATEMENT

Some forts are more difficult to climb than the others. Goku has read about all the forts and has given the i-th fort a difficulty level of Di. Note that these difficulty levels are relative to Goku's experience and expertise of hiking so it is possible for a fort to have a negative difficulty level (indicating that it is a very easy hike relative to Goku's expertise of hiking).

Goku is going to hoist his own flags on some of the forts. A fort is "marked" if there is a flag hoisted on it, or there is a flag hoisted on a fort that is directly connected to it by a road.

Goku can hoist as many or as few (even zero) flags as per his wish. Goku seeks help from you as you are one of the best coders in Algorythmia to find out the maximum possible sum of difficulty levels of "marked" forts that Goku can get.

INPUT FORMAT

The first line of the input gives the number of test cases, T. T test cases follow. Each test case begins with a line containing the integer N, the number of forts. The second line contains N integers. The i-th of these is Di, the difficulty level of the i-th fort.

Then, N-1 lines follow. The i-th line gives Fi and Gi, indicating the i-th road connects Fort Fi to Fort Gi.

OUTPUT FORMAT & CONSTRAINTS

For each test case, output the maximum possible sum of difficulty levels of "marked" forts Goku can get.

SAMPLE TESTCASES

(1) SAMPLE INPUT: 3

```
9
-10 4 -10 8 20 30 -2 -3 7
1 4
2 4
4 3
9 4
9 8
7 5
6 7
7 9
4
-2 20 20 20
1 2
1 3
1 4
5
-5 -10 8 -7 -2
5 4
4 3
3 2
2 1
```

SAMPLE OUTPUT: 67

```
58
0
```

SOLUTION

```cpp
#include <bits/stdc++.h>

using namespace std;

#define siz(x) ((int)(x).size())
#define all(x) (x).begin(),(x).end()
#define foreach(it,a) for(__typeof((a).begin()) it=(a).begin();it!=(a).end();it++)
#define rep(i,a,b) for (int i=(a),_ed=(b);i<_ed;i++)
#define per(i,a,b) for (int i=(b)-1,_ed=(a);i>=_ed;i--)
#define pb push_back
#define fi first
#define se second
#define mk(a,b)  make_pair((a), (b))
#define fastio \
  ios_base::sync_with_stdio(0); \
  cin.tie(0); \
  cout.tie(0);


typedef long long ll;
typedef unsigned long long ull;
typedef pair<int,int> pii;

const int maxN = 100000+13;

int b[maxN];
vector<int> e[maxN];
int isselected[maxN];

int calc(int n) {
  int ret = 0;
  for(int i = 1; i <= n; i++) {
    if(isselected[i]) {
      ret += b[i];
      continue;
    }
    for(auto tt : e[i]) {
      if(isselected[tt]) {
        ret += b[i];
        break;
      }
    }
  }
  return ret;
}
```

SOLUTION

```cpp
int main() {
  fastio;
  int T; cin >> T;
  for(int cas = 1; cas <= T; cas ++) {
    int N; cin >> N;
    for(int i = 1; i <= N; i++) {
      cin >> b[i];
      e[i].clear();
    }
    for(int i = 1; i < N; i++) {
      int x,y; cin >> x >> y;
      e[x].pb(y);
      e[y].pb(x);
    }
    int ans = 0;
    for(int i = 0; i < (1<<N); i++) {
      int tn = i;
      for(int j = 1; j <= N; j++) {
        isselected[j] = tn&1;
        tn >>=1;
      }
      int ta = calc(N);
      if(ta > ans) ans = ta;
    }
    cout << ans << endl;
  }


  return 0;
}
```