```cpp
//L2 004
#include <bits/stdc++.h>

using namespace std;
typedef long long ll;
typedef pair<int,int> pii;
const int mod = 1e9 + 7, N = 2e5 + 10;

int a[N], ans, tot;

void find1(int l, int r, int f){
    if(l > r || !ans) return;
    int mid = r;
    for(int i = l + 1; i <= r; i++){
        if(a[i] >= a[l]){
            mid = i - 1;
            break;
        }
    }
    for(int i = mid + 1; i <= r; i++){
        if(a[i] < a[l]){
            ans = 0;
            break;
        }
    }
    find1(l + 1, mid, f);
    find1(mid + 1,  r, f);
    if(f){
        if(tot) cout << ' ';
        cout << a[l];
        tot++;
    }
}

void find2(int l, int r, int f){
    if(l > r || !ans) return;
    int mid = r;
    for(int i = l + 1; i <= r; i++){
        if(a[i] < a[l]){
            mid = i - 1;
            break;
        }
    }
    for(int i = mid + 1; i <= r; i++){
        if(a[i] >= a[l]){
            ans = 0;
            break;
        }
    }
    find2(l + 1, mid, f);
    find2(mid + 1,  r, f);
    if(f){
        if(tot) cout << ' ';
        cout << a[l];
        tot++;
```

```
    }
}

void solve(){
    int n;
    cin >> n;
    for(int i = 1; i <= n; i++) cin >> a[i];
    ans = 1;
    find1(1, n, 0);
    if(ans){
        cout << "YES\n";
        find1(1, n, 1);
        return;
    }
    ans = 1;
    find2(1, n, 0);
    if(ans){
        cout << "YES\n";
        find2(1, n, 1);
        return;
    }
    cout << "NO\n";
}

int main(){
    int T = 1;
    ios::sync_with_stdio(false);
//    cin >> T;
    while(T--) solve();
    return 0;
}
```

```
//L2 006
#include <bits/stdc++.h>

using namespace std;
typedef long long ll;
typedef pair<int,int> pii;
const int mod = 1e9 + 7, N = 30 + 10;

vector<int> p[N];// p i 表示第i层的结点
int a[N], b[N];

void find(int l, int r, int l1, int r1, int depth){// l r 中序遍历的区间  l1 r1 后序遍历的区间  depth 当前深度
    if(l > r) return;
    p[depth].push_back(a[r1]);
    int mid;
    for(int i = l; i <= r; i++){
        if(b[i] == a[r1]){
            mid = i;
            break;
        }
```

```
    }
    // 左子树区间长度 mid - l
    find(l, mid - 1, l1, l1 + mid - l - 1, depth + 1);// 左子树
    find(mid + 1, r, l1 + mid - l, r1 - 1, depth + 1); // 右子树
}

void solve(){
    int n;
    cin >> n;
    for(int i = 1; i <= n; i++) cin >> a[i]; // 后序
    for(int i = 1; i <= n; i++) cin >> b[i]; // 中序
    int f = 1, cnt = 0;
    find(1, n, 1, n, 1);
    while(!p[f].empty()){
        for(int i : p[f]){
            if(cnt) cout << ' ';
            cout << i;
            cnt++;
        }
        f++;
    }
}

int main(){
    int T = 1;
    ios::sync_with_stdio(false);
//    cin >> T;
    while(T--) solve();
    return 0;
}
```

```
//L2 012
#include <bits/stdc++.h>

using namespace std;
typedef long long ll;
typedef pair<int,int> pii;
const int mod = 1e9 + 7, N = 2e5 + 10, M = 10000;

int tree[N], id[N];

void up(int k){
    while(k > 1){
        if(tree[k] < tree[k / 2]) swap(tree[k], tree[k / 2]);
        else break;
        k = k / 2;
    }
}

void solve(){
    int n, m;
    cin >> n >> m;
    for(int i = 1; i <= n; i++){
        cin >> tree[i];
        up(i);
```

```
    }
    for(int i = 1; i <= n; i++) id[tree[i] + M] = i;
    while(m--){
        string s;
        int a, b;
        cin >> a >> s;
        a += M;
        if(s == "and"){// 是否是兄弟结点
            cin >> b >> s >> s;
            b += M;
            if(id[a] / 2 == id[b] / 2){
                cout << "T\n";
            }else{
                cout << "F\n";
            }
        }else{
            cin >> s >> s;
            if(s == "root"){
                if(id[a] == 1){
                    cout << "T\n";
                }else{
                    cout << "F\n";
                }
            }else if(s == "child"){// a 是否是 b的孩子
                cin >> s >> b;
                b += M;
                if(id[a] / 2 == id[b]){
                    cout << "T\n";
                }else{
                    cout << "F\n";
                }
            }else{
                cin >> s >> b;
                b += M;
                if(id[b] / 2 == id[a]){
                    cout << "T\n";
                }else{
                    cout << "F\n";
                }
            }
        }
    }
}

int main(){
    int T = 1;
    ios::sync_with_stdio(false);
//    cin >> T;
    while(T--) solve();
    return 0;
}
```

```
//L2 035
#include <bits/stdc++.h>
```

```cpp
using namespace std;
typedef long long ll;
typedef pair<int,int> pii;
const int mod = 1e9 + 7, N = 2e5 + 10, M = 10000;

vector<int> p;
int ans[N];
void find(int k, int n){
    if(k > n) return;
    find(k * 2, n);
    find(k * 2 + 1, n);
    p.push_back(k);
}

void solve(){
    int n;
    cin >> n;
    find(1, n);
    for(int i = 0, x; i < n; i++){
        cin >> x;
        ans[p[i]] = x;
    }
    cout << ans[1];
    for(int i = 2; i <= n; i++) cout << " " << ans[i];
}

int main(){
    int T = 1;
    ios::sync_with_stdio(false);
//    cin >> T;
    while(T--) solve();
    return 0;
}
```