

```

#include <iostream>
#include <fstream>
#include "bintree_eda.h"

using namespace std;

template <class T >
void salvar(bintree<T> const& arbol, int &perdidos, int &grupos) {

    int perdidosR = 0, perdidosI = 0;

    if (arbol.left().empty() && arbol.right().empty()) {

        if (arbol.root() != 0) {

            grupos++;

        }

        perdidos += arbol.root();
    }
    else if (arbol.left().empty() || arbol.right().empty()) {

        if (arbol.left().empty()) salvar(arbol.right(), perdidosI, grupos);
        else salvar(arbol.left(), perdidosI, grupos);

        if (perdidosI == 0) {

            if (arbol.root() != 0) {

                grupos++;

            }

        }

        perdidos += arbol.root() + perdidosI;
    }
    else {

        salvar(arbol.left(), perdidosI, grupos);
        salvar(arbol.right(), perdidosR, grupos);

        if (perdidosI == 0 && perdidosR == 0) {

            if (arbol.root() != 0) {

                grupos++;

            }

        }

        perdidos += max(perdidosI, perdidosR) + arbol.root();
    }
}

```

```

void resuelveCaso() {

    bintree <int > arbol = leerArbol(-1);

    int perdidos = 0, grupos = 0;

    salvar(arbol, perdidos, grupos);

    cout << grupos << " " << perdidos << endl;
}

int main() {
    // Para la entrada por fichero.
    // Comentar para acepta el reto
#ifdef DOMJUDGE
    ifstream in("datos.txt");
    auto cinbuf = std::cin.rdbuf(in.rdbuf()); //save old buf and redirect std::cin to
casos.txt
#endif

    int numCasos;
    cin >> numCasos;
    for (int i = 0; i < numCasos; ++i)
        resuelveCaso();

    // Para restablecer entrada. Comentar para acepta el reto
#ifdef DOMJUDGE // para dejar todo como estaba al principio
    std::cin.rdbuf(cinbuf);
    system("PAUSE");
#endif

    return 0;
}

```