

1. Problema

Unir dos n árboles binarios utilizando el peso de los nodos.

2. Código

2.1. Función Union

```
void ArbolBinario::uni(list<ArbolBinario>& r){
    FibonacciHeap<NodoDT0> roots;
    for(ArbolBinario ab : r){
        roots.insert(NodoDT0(ab.root));
    }
    if(root){
        roots.insert(NodoDT0(root));
        root = nullptr;
    }
    return _uni(roots);
}

void ArbolBinario::_uni(FibonacciHeap<NodoDT0>& roots){
    if(roots.empty())return;
    if(root == nullptr){
        root = roots.popMin().nodo;
        return _uni(roots);
    }
    Nodo * menor = roots.popMin().nodo;
    Nodo * nuevo = new Nodo(root->valor + menor->valor);
    nuevo->hijos[0] = root;
    nuevo->hijos[1] = menor;
    root = nuevo;
    return _uni(roots);
}
```

2.2. Main

```
#include <iostream>
#include "ArbolBinario.h"

using namespace std;

int main()
{
    cout<<"INGRESE SU N"<<endl;
    int n;
    cin>>n;
    ArbolBinario arbolito;
    list<ArbolBinario> arbolitos;
    for(int i = 0; i < n; i++){
        cout<<"CUANTOS NODOS TENDRA EL ARBOL"<<i+1<<" ?? "<<endl;
        int n2;
        cin>>n2;
        ArbolBinario temp;
        for(int j = 0; j < n2; j++){
            cout<<"INGRESE EL NODO " <<j + 1<<endl;
            int val;
            cin>>val;
            temp.insert(val);
        }
        arbolitos.push_back(temp);
    }
    arbolito.uni(arbolitos);
    arbolito.print();
}
```

3. Ejemplo

Para el ejemplo ingreso un $n = 3$. El primer arbol, mirando por amplitud, es: 4,3,10. El segundo arbol: 6. El tercer arbol: 11,1,18,15.

```
ArbolesBinariosEjercicios
INGRESE SU N
3
CUANTOS NODOS TENDRA EL ARBOL1??
3
INGRESE EL NODO 1
4
INGRESE EL NODO 2
3
INGRESE EL NODO 3
12
CUANTOS NODOS TENDRA EL ARBOL2??
1
INGRESE EL NODO 1
6
CUANTOS NODOS TENDRA EL ARBOL3??
4
INGRESE EL NODO 1
11
INGRESE EL NODO 2
1
INGRESE EL NODO 3
18
INGRESE EL NODO 4
15
```

Figura 1: Consola del programa

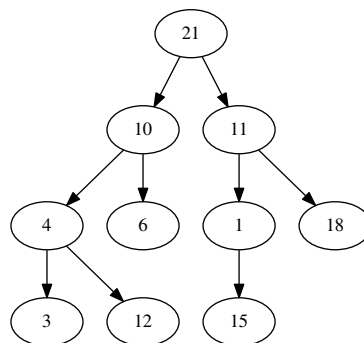


Figura 2: Arbol Resultante