ALEXANDRE SÁNCHEZ CASALS

```
#include <iostream>
#include "BinTree.hh"
#include <algorithm>
using namespace std;
```

* Pre: a = A, A no és buit */
/* Post: el resultat és la suma del subarbre amb suma màxima */

```
int i_suma_max_subarbre(BinTree<int>& a){

      if (a.empty()){
            return a.value();
            }
      else{
            int se,sd;
            se = sd = 0;
            se += (a.value()+i_suma_max_subarbre(a.left()));
            sd += (a.value()+i_suma_max_subarbre(a.right()));
            return max(se,sd);
            }
      }

       /*
      CAS BÀSIC:
            – Si a.empty() = true, per la pre, ens trobem en una fulla, i el subarbre major és ella mateixa.
      CASOS RECURSIUS:
            – Si no és fulla comprovem les sumes dels subarbres esquerra i dret i ens quedem amb la màxima d'ambdues.
      ACABAMENT:
            – Es manté dins mentres la crida no s'executi amb una fulla ( cas en que l'abre "a" retorni true en la crida de a.empty()).
      */

int suma_max_subarbre(BinTree<int>& a);
/* Pre: a = A, A no és buit */
/* Post: el resultat és la suma del subarbre no buit d'A amb suma màxima */
      int suma = i_suma_max_subarbre(a);
      return suma;
```