

ALEXANDRE SÁNCHEZ CASALS

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

```
bool es_complet(BinTree<int>& a)
```

```
/* Pre: a = A */
```

```
/* Post: el resultat indica si A és complet */
```

```
    if (a.left().empty() and a.right().empty()){
        return true;
    }
    else if (a.left().empty() and !a.right().empty()){
        return false;
    }
    else{
        bool e,d;
        e = es_complet(a.left());
        d = es_complet(a.right());
        if (e and d){
            return true;
        }
        return false;
    }
}
```

```
void completa(BinTree<int> &a, int h, int x)
```

```
/* Pre: a = A, h>=altura(A) */
```

```
/* Post: a és l'arbre A completat fins a un arbre complet d'altura h,
```

```
afegint-hi nodes amb valor x */
```

```
if (!a.es_complet()){
    int h2 = 1;
    bool bcomplet = false;
    int complet = 1;
    while (h2 <= h){
        while (!bcomplet){
            complet *= h2;
            if (!a.left().es_complet()){
                BinTree<int> a (x)
                //afegim nodes
            }
        }
    }
}
```

```
}
```

No tinc clar que es vagi a basar en el principi d inducció. No sé com procedir. La v2 serà correcte.

```
/* CAS BÀSIC:
```

```
-
```

```
CASOS RECURSIUS:
```

```
-
```

```
ACABAMENT:
```

```
-
```

```
*/
```