

Esercizio 2:

```
type TernaryTree = Leaf | TT of int * TernaryTree * TernaryTree * TernaryTree

let t = TT (3,TT (4,Leaf,Leaf,TT(3,Leaf,Leaf,Leaf)),TT (2,Leaf,Leaf,Leaf),Leaf)

// 3
// |-->4
//   |--> *
//   |--> *
//   |-->3
//       |--> *
//       |--> *
//       |--> *
// |-->2
//   |--> *
//   |--> *
//   |--> *
// |--> *
```

```
let rec prod x =
  match x with
  | Leaf -> 1
  | TT (n,t1,t2,t3) -> n * (prod t1) * (prod t2) * (prod t3)
```