

Esercizio 11.4. Si considerino le seguenti dichiarazioni e definizioni:

```
class Nodo {
private:
    Nodo(string st="***", Nodo* s=0, Nodo* d=0): info(st), sx(s),
                                                dx(d) {}

    string info;
    Nodo* sx;
    Nodo* dx;
};

class Albero {
public:
    Albero(): radice(0) {}
    Albero(const Albero&); // dichiarazione costruttore di copia
private:
    Nodo* radice;
};
```

Quindi, gli oggetti della classe Albero rappresentano *alberi binari ricorsivamente definiti di stringhe*. Si ridefinisca il costruttore di copia di Albero in modo che esegua copie profonde.

Soluzione (Possibile) Esercizio 11.4.

```
class Nodo {
friend class Albero;
private:
    Nodo(string st="***", Nodo* s=0, Nodo* d=0): info(st), sx(s),
                                                dx(d) {}

    string info;

    Nodo* sx;
    Nodo* dx;
};

class Albero {
public:
    Albero(): radice(0) {}
    Albero(const Albero& t) {radice = copia(t.radice);}
private:
    Nodo* radice;
    static Nodo* copia(Nodo* p) {
        if (!p) return 0;
        else return new Nodo(p->info, copia(p->sx), copia(p->dx));
    }
};
```