

**1.** Используя интеллектуальные указатели типа `shared_ptr`, реализуйте дерево, в котором каждый узел имеет указатель на своего родителя и двух потомков.

*Пример класса — узла дерева.*

```
class A: public std::enable_shared_from_this<A>
{
public:
    void addChildren(std::shared_ptr<A> &lChild, std::shared_ptr<A> &rChild)
    {
        leftChild = lChild;
        rightChild = rChild;
        leftChild->setParent(shared_from_this());
        rightChild->setParent(shared_from_this());
    }
    void setParent(std::shared_ptr<A> parent)
    {
        this->parent = parent;
    }
private:
    std::shared_ptr<A> leftChild; // указатель на левый узел-потомок
    std::shared_ptr<A> rightChild; // указатель на левый узел-потомок
    std::shared_ptr<A> parent; // указатель на узел-родитель
};
```

*Наследование от `std::enable_shared_from_this<A>` нужно для того, чтобы можно было использовать `shared_from_this()`.*

**2.** Реализуйте алгоритм бинарного поиска в отсортированном контейнере `vector` с помощью итераторов.