

Решение задачи можно присылать (предпочтительнее ссылка на github, и т.п.) на [vkonoovodov@gmail.com](mailto:vkonoovodov@gmail.com). Бонусные баллы за решения задачи могут быть поставлены только нескольким первым приславшим правильное решение. Решение должно быть оформлено в виде компилирующегося кода.

**Внимание!** В тексте письма напишите  $N$  пунктов про каждое ваше исправление кода словами: что нужно сделать, чтобы исправить каждую ошибку (про каждую ошибку — отдельный пункт).

## Задача

В файле `tree.h` приведена реализация двоичного дерева, заданного типом `TNode`, который кроме значения основного типа хранит указатели на левого и правого потомков, а также указатель на родителя. В интерфейсе определены обычным образом `get`-методы, метод `createLeaf` для создания дерева из одного листа, методы `replaceLeft` и `replaceRight` для замены левого и правого потомка соответственно, метод `fork` для создания вершины с заданными потомками, методы `removeLeft` и `removeRight` для удаления потомков.

Найдите ошибки, связанные с работой с памятью, и исправьте их. Исправьте код так, чтобы он компилировался. Объясните ваши исправления.

Не допускается изменение интерфейса работы с классом (например, добавление новых методов создания вершин), при этом разрешается менять аргументы методов класса `TNode` (без изменения семантики самих методов).

В файле `tree.cpp` приведен пример работы с классом `TNode`.