

Реализуйте двоичное дерево поиска в виде шаблонного класса на языке C++. При реализации ориентируйтесь на интерфейс контейнера [`std::set<Key, Compare>`](#) из стандартной библиотеки (вместо итераторов допустимо использовать указатели на непосредственно узлы). В структуре узла обязан быть указатель на родительский узел.

1. Реализовать операции:
  1. Добавление элемента.
  2. Поиск элемента.
  3. Поиск минимального и максимального элементов.
  4. Удаление элементов.
  5. Поиск первого элемента, меньше или равного заданному и первого больше заданного (см. семантику `lower_bound/upper_bound` из стандартной библиотеки).
  6. Вывод на экран:
    1. Левый/Текущий/Правый, с использованием рекурсии.
    2. Правый/Текущий/Левый, с использованием стека и без рекурсии.
    3. По слоям, с использованием очереди.
  7. Сравнение двух деревьев на равенство с учётом структуры, рекурсивно.
  8. Поиск для заданного (в виде указателя, а лучше итератора) элемента следующего и предыдущего элементов (два разных метода). Учтите, что следующим элементом не всегда является непосредственный потомок заданного узла. Поведение должно быть аналогичным оператору `++` у итератора для стандартного `set`.
2. Реализуйте алгоритм "Решето Эратосфена" двумя разными способами и сравните эффективность:
  1. На основе вашего бинарного дерева поиска.
  2. На основе массива.