ВШЭ АиСД 2021. Сбалансированные деревья

4 дек 2021, 18:44:15
старт: 29 окт 2021, 12:00:00
финиш: 9 ноя 2021, 00:00:01
длительность: 10д. 12ч.
начало: 29 окт 2021, 12:00:00
конец: 9 ноя 2021, 00:00:01

D. Просто КЧД (0.4)

Ограничение времени	4 секунды
Ограничение памяти	64.0 Мб
Ввод	стандартный ввод
Вывод	стандартный вывод

В этой задаче вам предстоит реализовать классическое **красно-черное дерево**. Требования к асимптотической сложности операций такие же, как у стандартного контейнера std::set (ссылка), если не указано иное.

Вы должны написать шаблон RBTree, который параметризуется типом хранящихся элементов, иными словами, следующее:

```
template <class ValueType>
class RBTree {
// ваша реализация
};
```

Ваш класс должен содержать следующие конструкторы и методы:

1. Конструктор по умолчанию.
2. Конструктор копирования (ссылка).
3. Оператор присваивания.
4. Деструктор.
5. Константный метод `size`, возвращающий количество элементов в дереве.
6. Константный метод `empty`, возвращающий true, если дерево пустое, и false в противном случае.
7. Метод `insert`, который добавляет элемент в дерево. Если элемент уже присутствует, метод не должен ничего делать. Тип возвращаемого значения void.
8. Метод `erase`, который удаляет переданный элемент из дерева. Если искомого элемента в дереве нет, метод не должен ничего делать. Тип возвращаемого значения void.
9. Константный метод `find`, возвращающий указатель на значение, соответствующее элементу дерева, если оно в нем присутствует, и nullptr в противном случае.
10. Константный метод `lower_bound`, который по заданному значению x возвращает указатель на минимальное значение y, соответствующее элементу дерева, такое что $y \geq x$, или nullptr, если искомого элемента в дереве нет.
11. Константный метод `traversal`, который возвращает массив вершин по возрастанию их значений. Тип возвращаемого значения `ValueType *`.

Особое внимание обратите на аккуратную работу с памятью. Это касается конструктора копирования, оператора присваивания и деструктора. Копирование деревьев подразумевает полное ("глубокое") копирование всех узлов, а не просто копирование корня. В деструкторе вы должны освобождать всю выделенную память.

Обратите внимание, название метода `lower_bound` не соответствует действующему кодстайлу. Если линтер начнет ругаться, то добавьте строчку `// NOLINT` в конце строчки - объявления метода.

Для сравнения элементов используйте только оператор <.

Примечания

Вы должны прислать .cpp файл, содержащий определение вашего класса. Ваш класс должен примерно иметь следующий интерфейс: ссылка. Вам **не нужно** писать реализацию конкретно данного заголовочного файла. Можно добавлять свои приватные поля / методы в этот класс.

Обратите внимание, что тестирующая программа достаточно строго проверяет требования из условия. В случае их несоблюдения вы будете получать вердикты.

1. PCF (несоответствие стилю / неправильное название файла / использование стандартных контейнеров, запрещенных в условии - подробное описание смотрите в отчете)
2. CE (несоответствие стилю (подробное описание смотрите в отчете) / неверные сигнатуры методов / переопределение `main` / другие ошибки компиляции).
3. RE (memory leak / undefined behaviour / прочие проблемы с памятью - пишите юнит-тесты и прогоняйте решение с санитайзерами)
4. WA (неправильный ответ - пишите юнит-тесты)
5. TL (превышение времени - пишите юнит-тесты)
6. ML (превышение памяти)

Рекомендую написать вначале программу, которая ничего не делает, но хотя бы компилируется. И только после добавлять постепенно функционал, смотреть на результаты тестов. Похоже немного на TDD, но здесь тесты уже за вас написаны.

by Игумнов Никита

Ограничения

1. Использование других видов деревьев вместо красно-черного запрещено.
2. *Кому нечем заняться*: напишите еще свой класс `iterator`, соответствующий итератору, с помощью которого можно просмотреть содержимое контейнера, а также соответствующие методы `begin` и `end`.

Система оценки

Группа	Баллы	Необх. группы	Комментарий
0	6	–	Небольшие тесты, проверяющие выполнение основных требований интерфейса и функционала класса в задаче.
1	2	0	**Offline-проверка**

Набрать здесь | Отправить файл

```
1
```

Отправить

Предыдущая