ВШЭ АиСД 2021. B, B+, Splay деревья

4 дек 2021, 18:41:34
старт: 8 ноя 2021, 20:00:00
финиш: 18 ноя 2021, 12:00:00
длительность: 9д. 16ч.
начало: 8 ноя 2021, 20:00:00
конец: 18 ноя 2021, 12:00:00

C. Splay-поиск (0.3)

Ограничение времени	1 секунда
Ограничение памяти	64.0 Мб
Ввод	стандартный ввод
Вывод	стандартный вывод

В этой задаче Вам требуется написать свой класс - расширение обычного бинарного дерева поиска (хранит целочисленные ключи), чтобы быстрее выполнять **поиск значений, которые использовались недавно**.

Дерево создается по входной последовательности целых чисел длины N.

Реализуемый Вами класс `SplayTree` должен содержать следующие публичные методы:

1. Конструктор без параметра.
2. Метод `insert(int key)`, который добавляет элемент в дерево **без балансировки**.
Если элемент уже присутствует, метод не должен ничего делать.
Тип возвращаемого значения `void`.
Реализация - итерационная.
3. Константный метод `find(int key)`, возвращающий указатель на вершину дерева, в которой хранится значение `key`, если оно в нем присутствует, и `nullptr` в противном случае.
Реализация - итерационная.
4. Метод `splay(Node *node)`, разворачивающий дерево и возвращающий суммарное количество поворотов дерева, которые потребовалось выполнить, чтобы найденная вершина стала корнем дерева. При этом, сложные повороты дерева, состоящие из двух *разных* поворотов, т. е. «zig-zag» или «zag-zig» считаются одним поворотом при подсчете.
5. Константный метод `getHeight()`, возвращающий высоту дерева. Высота дерева соответствует длине самого длинного пути из корня дерева в его листья. Дерево, состоящее из одной вершины, имеет высоту 0.

Примечания

Вы должны загрузить `.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.

Гарантируется, что

- длина N входной числовой последовательности находится в пределах $0 \le N \le 2 \cdot 10^4$ (числа по модулю не больше 10^9),
- количество запросов q находится в пределах $1 \le q \le 2 \cdot 10^6$ (каждый запрос содержит число `key` (по модулю не больше 10^9); при обработке каждого запроса единожды вызываются методы `find(key)` и `getHeight()`).

Система оценки

Группа	Баллы	Доп. ограничения		Необх. группы	Комментарий
		N	q		
0	1	–	–	–	Компиляция.
1	2	$N \le 5$	$q \le 5$	0	
2	2	$N \le 200$	$q \le 2 \cdot 10^3$	0	
3	3	$N \le 2 \cdot 10^3$	$q \le 2 \cdot 10^4$	0	

Группа	Баллы	Доп. ограничения		Необх. группы	Комментарий
4	2	$N \leq 2 \cdot 10^4$	$q \leq 2 \cdot 10^6$	0 – 3	**Offline-проверка**

Набрать здесь | Отправить файл

```cpp
#include <iostream>

struct Node {
public:
    int key;
    Node *parent;
    Node *leftChildren;
    Node *rightChildren;

    explicit Node(int value) {
        key = value;
        parent = nullptr;
        leftChildren = nullptr;
        rightChildren = nullptr;
    }
};

class SplayTree {
public:
    SplayTree() {
        root_ = nullptr;
    }

    ~SplayTree() {
        deleteNode(root_);
    }

    void insert(int key) {
        Node *node = new Node(key);
        Node *parent = nullptr;
        Node *root = root_;

        while (root) {
            parent = root;
            if (node->key < root->key) {
                root = root->leftChildren;
            } else if (node->key > root->key) {
                root = root->rightChildren;
```

Отправить

Предыдущая