

Отчет по сравнению CAVLTree на MemoryManager и new/delete.

Цель исследования:

Сравнить производительность CAVLTree при использовании двух разных методов управления памятью: MemoryManager и операторы new/delete.

Методология:

Для проведения сравнения были выполнены следующие шаги:

Реализованы две версии CAVLTree: одна использует MemoryManager ([/isamidinova-EDSA/avltrees_comparison/avltree/avltree.h](#)), вторая - операторы new/delete для управления памятью

([/isamidinova-EDSA/avltrees_comparison/avltree_new_delete/avl_new_delete.h](#)).

Для каждой версии были проведены четыре основных операции: добавление (add), поиск (find), обновление (update) и удаление (remove) элементов.

Каждая операция была выполнена миллион раз для сбора статистики о времени выполнения.

Время выполнения каждой операции было записано в соответствующий файл и на основе этих данных был сгенерирован график.

Результаты:

Измерения времени выполнения операций для каждой версии CAVLTree

представлены в файле `"timer_for_graph.txt"` для MemoryManager и

`"timer_for_graph_new_delete.txt"` для new/delete. Программа в файле

`"/isamidinova-EDSA/avltrees_comparison/graph/graph.py"` считывает данные из

файла и генерирует график в виде пар гистограмм для каждой операции

(пара - CAVLTree на основе MemoryManager и new/delete). Для воспроизведения

эксперимента можно зайти в папку `avltrees_comparison` и запустить

bash-скрипт `run.sh`. Он соберет проект, запустит оба исполняемых файла с

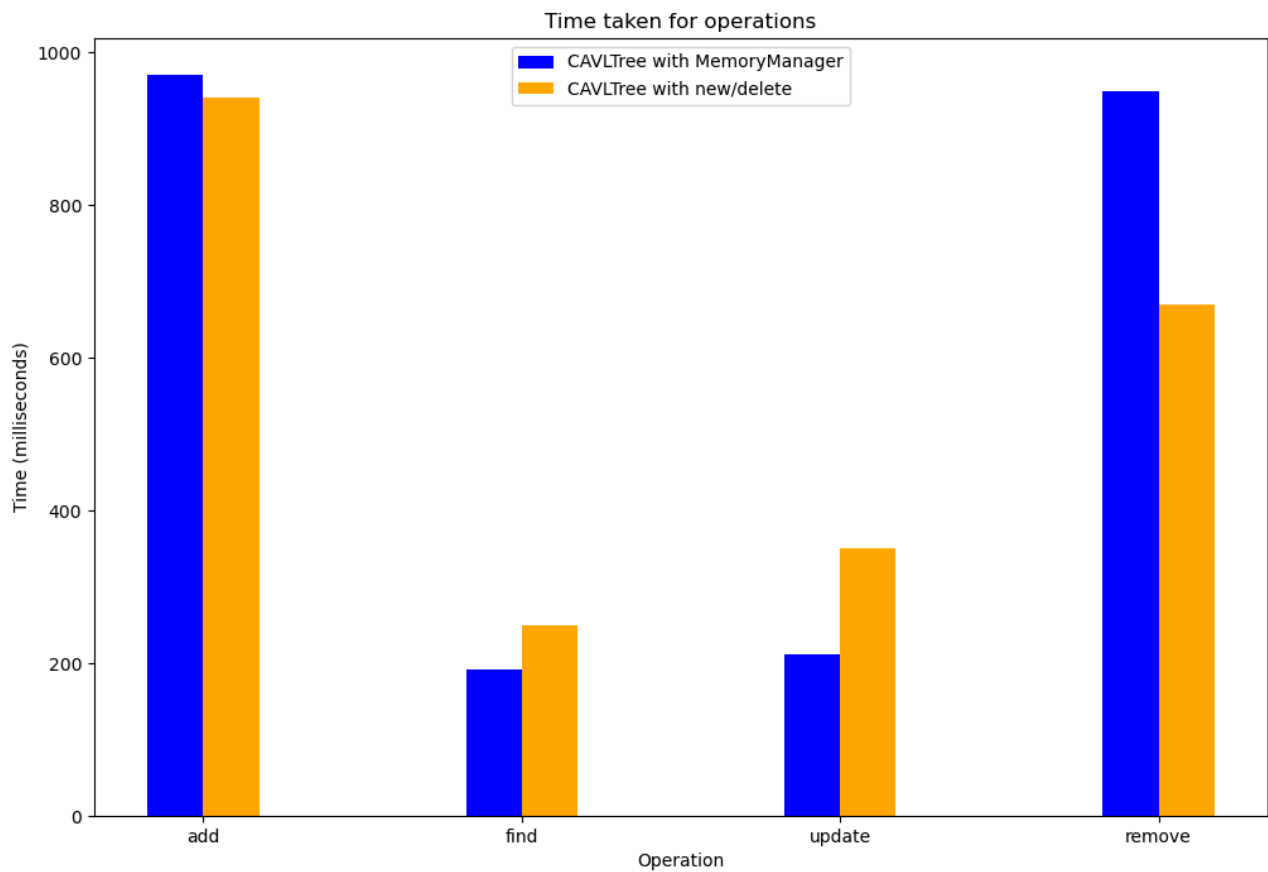
тестированием деревьев и выполнит python-скрипт по генерации графика.

Анализ:

На момент составления отчета результаты в миллисекундах:

CAVLTree с	MemoryManager	new/delete
Добавление(add)	970	941
Поиск(find)	191	250
Обновление(update)	212	351
Удаление(remove)	949	670

Ниже график на основе этих данных:



Из результатов видно, что в общем случае CAVLTree с использованием new/delete показывает лучшую производительность по сравнению с реализацией, использующей MemoryManager. Особенно это заметно в операциях добавления и удаления, где время выполнения с использованием new/delete оказалось меньше, чем с использованием MemoryManager.