Лабораторная работа №3 по курсу дискрeтного анализа:

исследование качества программ.

Выполнил студент группы М80-208Б-20 Зубко Дмитрий Валерьевич.

Условие

Для реализации словаря из предыдущей лабораторной работы, необходимо провести исследование скорости выполнения и потребления оперативной памяти. В случае выявления ошибок или явных недочетов, требуется их исправить.

# Дневник выполнения работы

В данной лабораторной работе я воспользовался утилитами: **gprof** и **valgrind**.

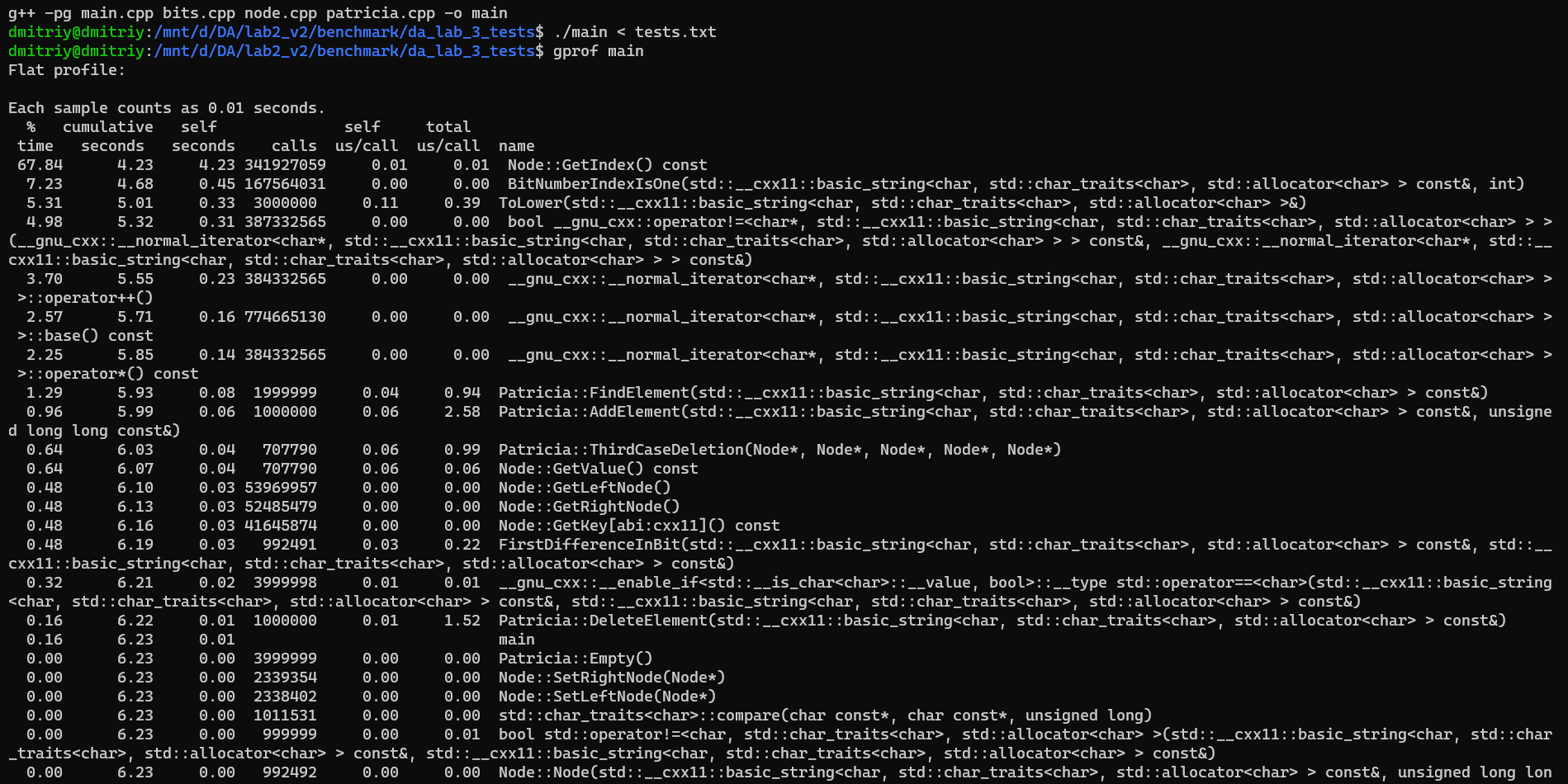
1. **Gprof –** позволяет получить данные профилирования программы. То есть, эта утилита позволяет измерить время работы всех функций, методов и операторов программы, количество их вызовов, долю от общего времени работы программы.

**Запуск:**

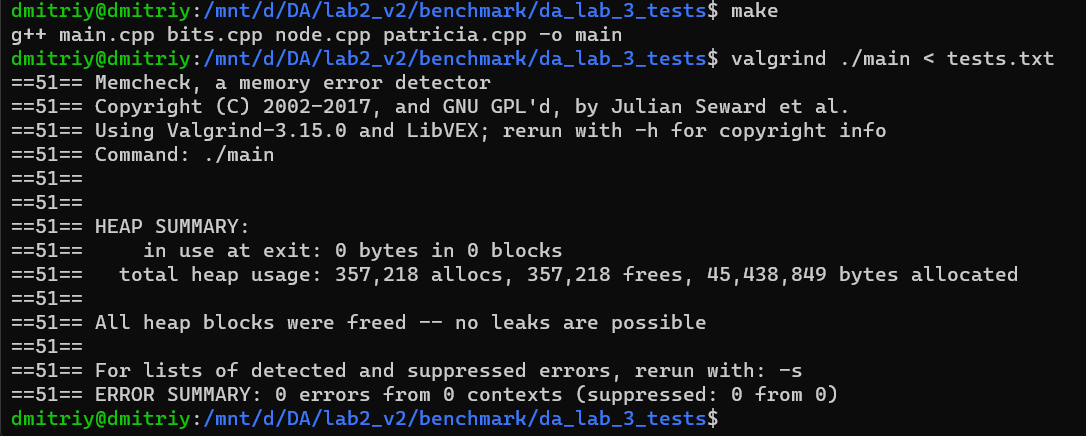
1. Генерация тестов: **python main.py 1000000**
2. Компиляция программы: **g++ -pg main.cpp bits.cpp node.cpp patricia.cpp -o main**
3. Запуск программы: **./main < test.txt**
4. Просмотр файла gmon.out, который является выводом нашей утилиты:

**gprof main**

Результат вывода утилиты **gprof** следующий:



Все остальные функции по данным gprof работали 0% времени, поэтому на скриншоте они отсутствуют. Больше всего времени заняла функция получения индекса, так как она вызывается чаще всего. Дальше идут функции для работы со строками: BitNumberIndexIsOne, ToLower.

1. **Valgrind** – мощное средство для поиска ошибок работы с памятью. Для проверки программы на ошибки с памятью нужно ввести в терминал:
2. Генерация тестов: **python main.py 1000000**
3. Компиляция программы: **g++ main.cpp bits.cpp node.cpp patricia.cpp -o main**
4. Valgrind ./main < tests.txt

Из вывода утилиты видно, что утечек памяти нет.

Вывод о найденных недочетах

Valgrind показал still reachable при запуске программы в первый раз. Чтобы это исправить я убрал из программы “строки-ускорители” std::ios::sync\_with\_stdio(false); std::cin.tie(nullptr); std::cout.tie(nullptr).

# Общие выводы о выполнении лабораторной работы

Данная лабораторная работа познакомила меня с утилитой gprof, помогла мне закрепить навыки работы с утилитой для нахождения утечек памяти valgrind. Я проанализировал время выполнения функций программы. Нашёл утечки памяти и устранил их. В дальнейшем при написании программ я буду стараться использовать эти утилиты чаще.