Отчет о выполнении задания 2  
«Бинарные деревья поиска и хеш-таблицы»

**Меняйлов Владислав Олегович**

Группа ИС-641

emindanew@yandex.ru

# Описание заданий

В задании 1 требовалось сравнить эффективность поиска элементов в бинарном дереве поиска и хэш-таблице.

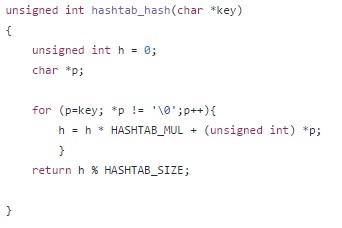
# **Двоичное дерево поиска это** [двоичное дерево](https://ru.wikipedia.org/wiki/Двоичное_дерево), для которого выполняются следующие дополнительные условия:

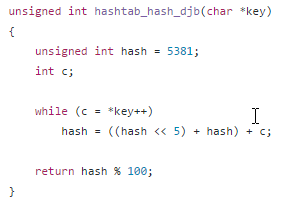
* Оба поддерева— левое и правое— являются двоичными деревьями поиска.
* У всех узлов *левого* поддерева произвольного узла X значения ключей данных *меньше*, нежели значение ключа данных самого узла X.
* У всех узлов *правого* поддерева произвольного узла X значения ключей данных *больше либо равно*, нежели значение ключа данных самого узла X.

Хеш-табли́ца — это [структура данных](https://ru.wikipedia.org/wiki/Структура_данных), реализующая интерфейс [ассоциативного массива](https://ru.wikipedia.org/wiki/Ассоциативный_массив), а именно, она позволяет хранить пары (ключ, значение) и выполнять три операции: операцию добавления новой пары, операцию поиска и операцию удаления пары по ключу.

В задании 3 требовалось сравнить эффективности поиска элементов в бинарном дереве и хеш-таблицу.

В задании 6 требовалось сравнить две хэш-функции djb и KP.





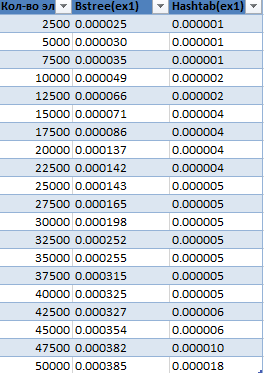
# Организация экспериментов

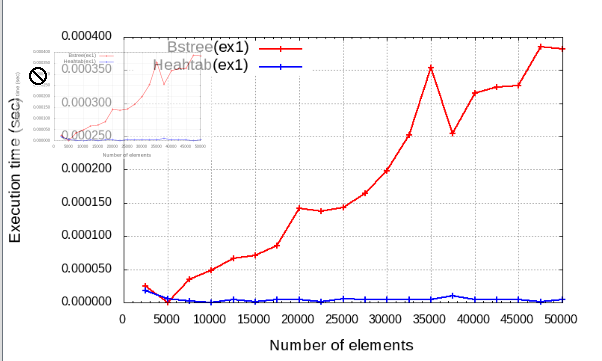
* Эксперименты проводились на ноутбуке Samsung   
  (Intel(R) Core(TM) i3-3110M CPU @2.40GHz 2.40GHz Ozu 4gb
* Операционная система Opensuze 20 x86\_64 (компилятор gcc 4.8.2)
* Ключи компиляции программы (см. README): -Wall -O2

# Результаты экспериментов

Эксперимент 1:

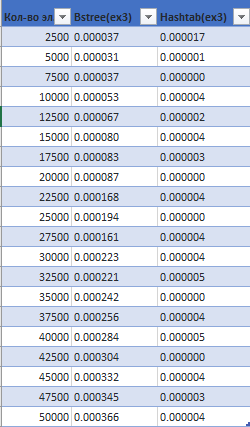
Эксперимент 1. Сравнение эффективности поиска элементов в бинарном дереве поиска и хеш-таблице в среднем случае. Из результатов экспериментов видно, что поиск в бинарном дереве работает дольше поиска в хеш-таблице.

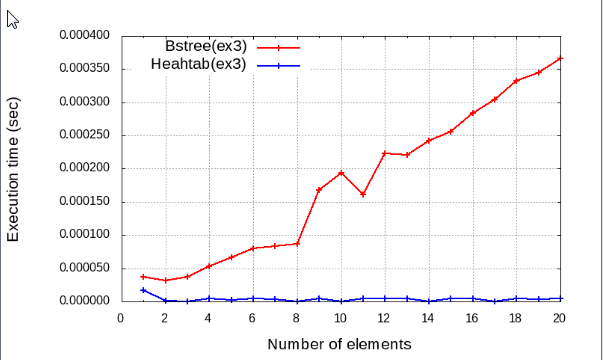




Эксперимент 3:

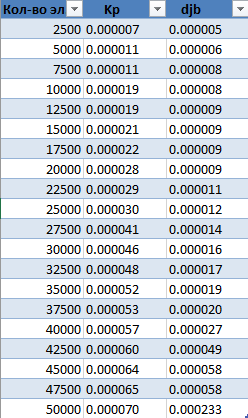
В задании 3 требовалось сравнить эффективности поиска элементов в бинарном дереве и хеш-таблицу.

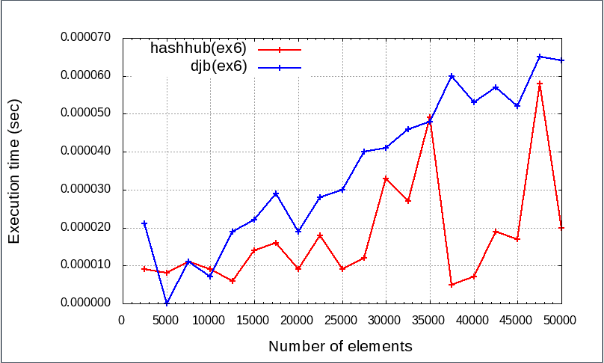




Эксперимент 6 :

Анализ эффективности хеш-функций. Kp and djb





# Контрольные вопросы

Словарь (ассоциативный массив, associative array, map, dictionary) – структура (контейнер) данных для хранения пар вида “ключ – значение” (key – value).

Двоичное дерево (Binary tree) – это дерево (структура данных), в которой каждый узел (node) имеет не более двух дочерних узлов (child nodes).

|  |  |  |
| --- | --- | --- |
| Операция | Средний случай | Худший случай |
| Добавление | O(log n) | O(n) |
| Поиск | O(log n) | O(n) |

Хеш-таблица (Hash table) – это структура данных для хранения пар “ключ – значение”

|  |  |  |
| --- | --- | --- |
| Операция | Средний случай | Худший случай |
| Добавление | O(1) | O(1) |
| Поиск | O(1+n/h) | O(n) |
| Удаление | O(1+n/h) | O(n) |

Хеш-функция (Hash function) – это функция преобразующая значения ключа (например: строки, числа, файла) в целое число. Значение возвращаемое хеш-функцией называется хеш-кодом.

Хорошая хеш-функция должна удовлетворять двум требованиям: ее вычисление должно быть очень быстрым и она должна минимизировать число коллизий.

Большинство первых работ, описывающих хеширование, посвящено методам борьбы с коллизиями в хеш-таблицах. Тогда хеш-функции применялись при поиске текста в файлах большого размера. Существует два основных метода борьбы с коллизиями в хеш-таблицах:

1) метод цепочек (метод прямого связывания);

2) метод открытой адресации.

При использовании метода цепочек в хеш-таблице хранятся пары «связный список ключей» — «хеш-код». Для каждого ключа хеш-функцией вычисляется хеш-код; если хеш-код был получен ранее (для другого ключа), ключ добавляется в существующий список ключей, парный хеш-коду; иначе, создаётся новая пара «список ключей» — «хеш-код», и ключ добавляется в созданный список. В общем случае, если имеется N ключей и M списков, средний размер хеш-таблицы состави. В этом случае при поиске по таблице по сравнению со случаем, в котором поиск выполняется последовательно, средний объём работ уменьшится примерно в M раз.

При использовании метода открытой адресации в хеш-таблице хранятся пары «ключ» — «хеш-код». Для каждого ключа хеш-функцией вычисляется хеш-код; пара «ключ» — «хеш-код» сохраняется в таблице. В этом случае при поиске по таблице по сравнению со случаем, в котором используются связные списки, ссылки не используются, выполняется последовательный перебор пар «ключ» — «хеш-код», перебор прекращается после обнаружения нужного ключа. Последовательность, в которой просматриваются ячейки таблицы, называется последовательностью проб.