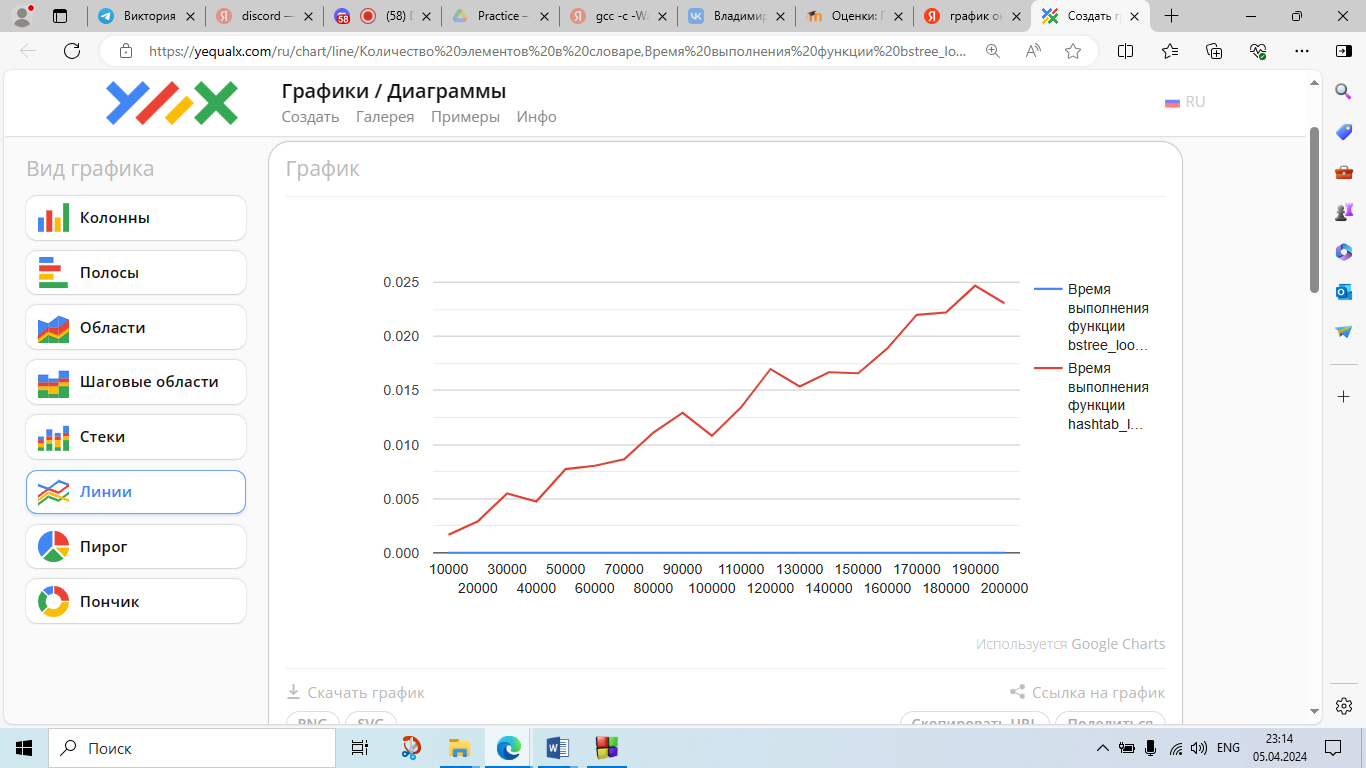
САОД №2

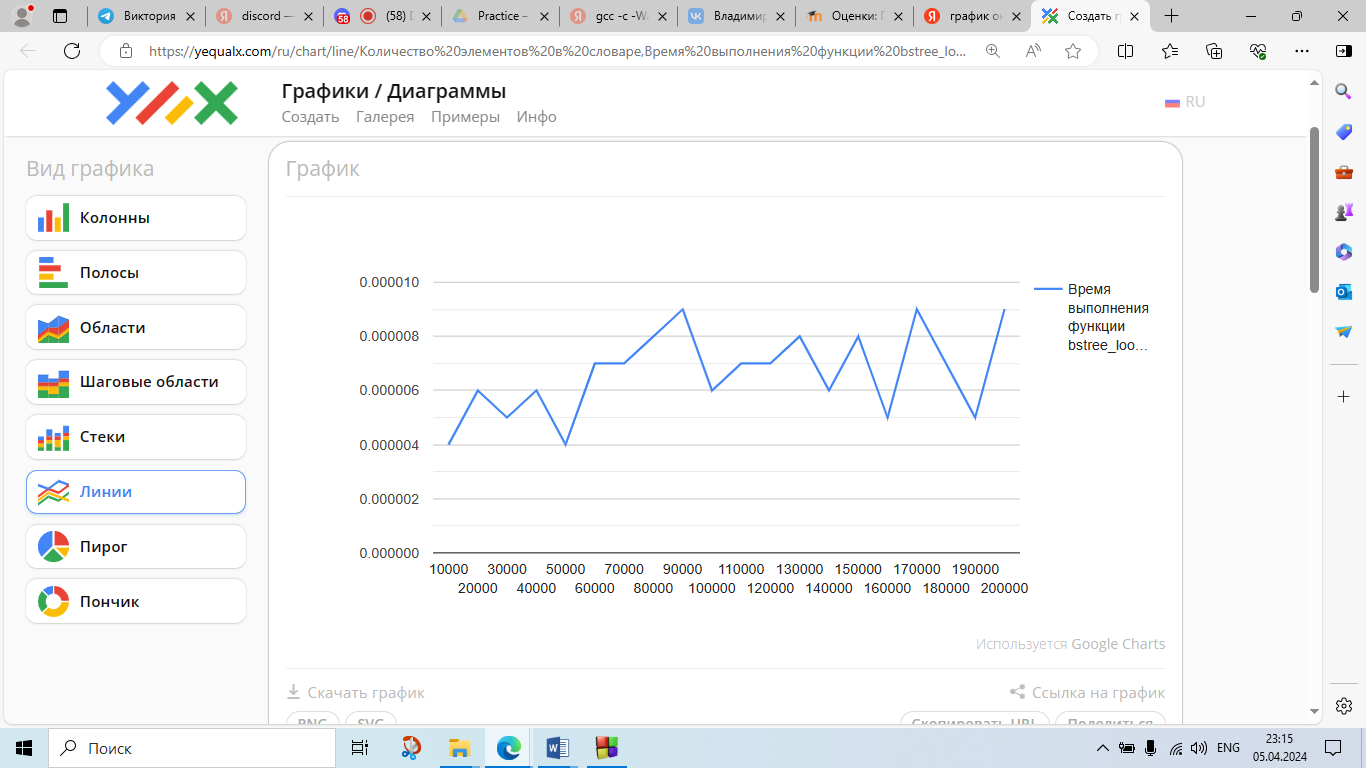
Вариант 4

ИА-332

1. Эксперимент 1

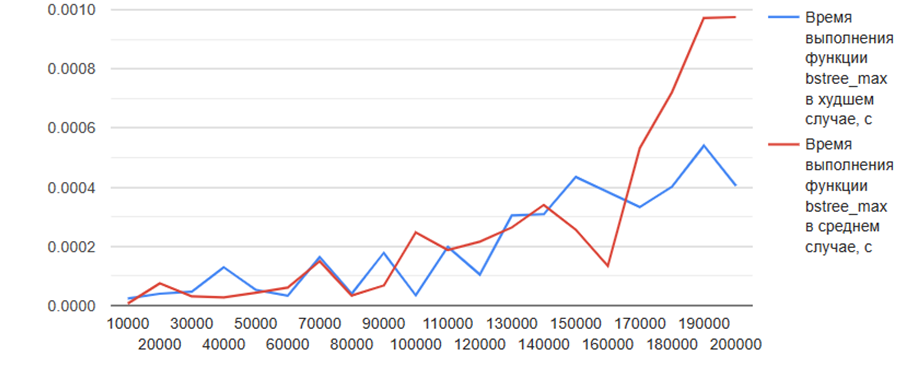
|  |  |  |  |
| --- | --- | --- | --- |
| # | Количество элементов в словаре | Время выполнения функции bstree\_lookup, с | Время выполнения функции hashtab\_lookup, с |
| 1 | 10000 | 0.000004 | 0.001665 |
| 2 | 20000 | 0.000006 | 0.002905 |
| 3 | 30000 | 0.000005 | 0.005493 |
| 4 | 40000 | 0.000006 | 0.004750 |
| 5 | 50000 | 0.000004 | 0.007739 |
| 6 | 60000 | 0.000007 | 0.008049 |
| 7 | 70000 | 0.000007 | 0.008638 |
| 8 | 80000 | 0.000008 | 0.011107 |
| 9 | 90000 | 0.000009 | 0.012934 |
| 10 | 100000 | 0.000006 | 0.010825 |
| 11 | 110000 | 0.000007 | 0.013470 |
| 12 | 120000 | 0.000007 | 0.016977 |
| 13 | 130000 | 0.000008 | 0.015363 |
| 14 | 140000 | 0.000006 | 0.016685 |
| 15 | 150000 | 0.000008 | 0.016592 |
| 16 | 160000 | 0.000005 | 0.018897 |
| 17 | 170000 | 0.000009 | 0.021984 |
| 18 | 180000 | 0.000007 | 0.022205 |
| 19 | 190000 | 0.000005 | 0.024679 |
| 20 | 200000 | 0.000009 | 0.023064 |





1. Эксперимент 5

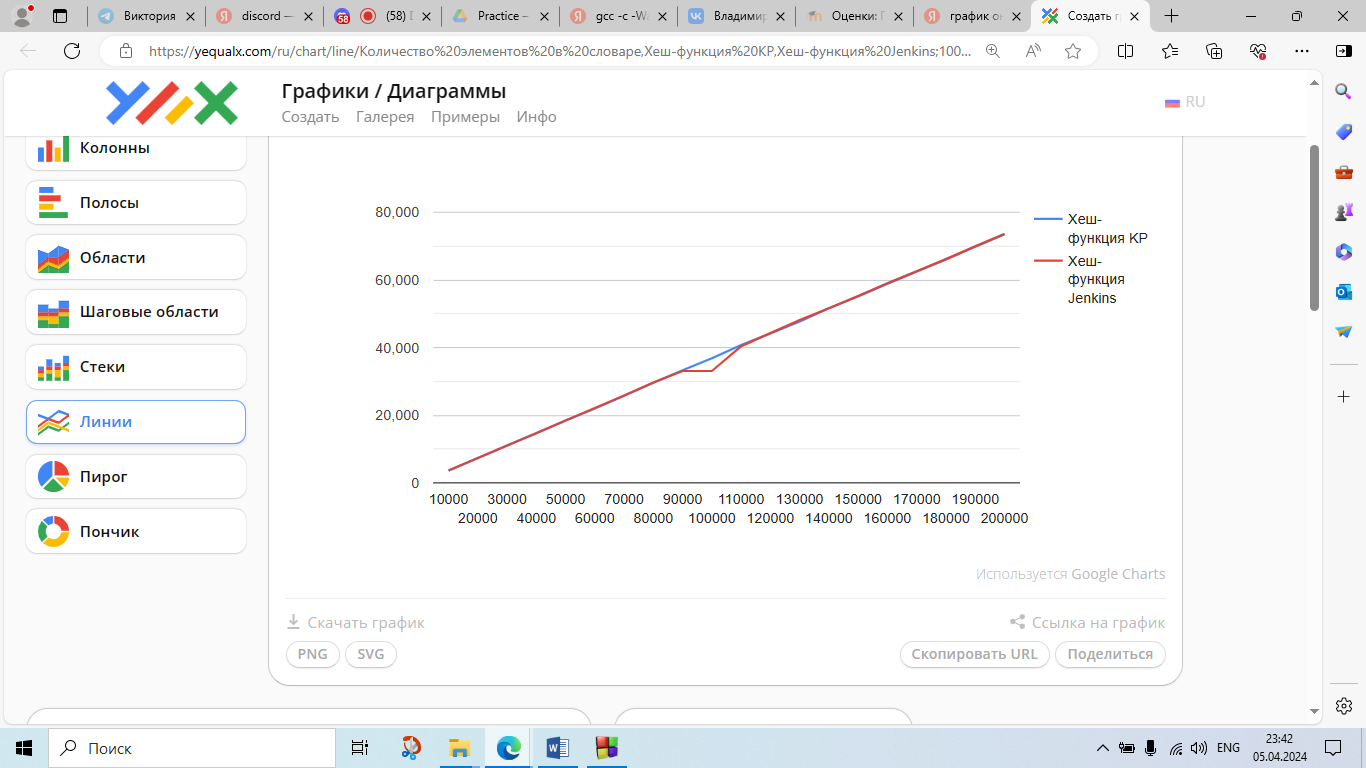
|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| # | Количество элементов в словаре | Время выполнения функции bstree\_max в худшем случае, с | Время выполнения функции bstree\_max в среднем случае, с |
| 1 | 10000 | 0.000024 | 0.000007 |
| 2 | 20000 | 0.000040 | 0.000075 |
| 3 | 30000 | 0.000047 | 0.000031 |
| 4 | 40000 | 0.000129 | 0.000028 |
| 5 | 50000 | 0.000053 | 0.000043 |
| 6 | 60000 | 0.000033 | 0.000061 |
| 7 | 70000 | 0.000164 | 0.000150 |
| 8 | 80000 | 0.000040 | 0.000034 |
| 9 | 90000 | 0.000178 | 0.000068 |
| 10 | 100000 | 0.000035 | 0.000247 |
| 11 | 110000 | 0.000198 | 0.000188 |
| 12 | 120000 | 0.000105 | 0.000216 |
| 13 | 130000 | 0.000305 | 0.000264 |
| 14 | 140000 | 0.000309 | 0.000340 |
| 15 | 150000 | 0.000435 | 0.000256 |
| 16 | 160000 | 0.000384 | 0.000134 |
| 17 | 170000 | 0.000333 | 0.000533 |
| 18 | 180000 | 0.000402 | 0.000720 |
| 19 | 190000 | 0.000541 | 0.000972 |
| 20 | 200000 | 0.000405 | 0.000975 |



6) Эксперимент 6 — хеш-функции КР, Jenkins

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Количество элементов в словаре | Хеш-функция KP | | Хеш-функция Jenkins | |
| Время выполнения  функции  hashtab\_lookup, с | Число  коллизий | Время выполнения  функции  hashtab\_lookup, с | Число  коллизий |
| 1 | 10000 | 0.001436 | 3681 | 0.0015599 | 3654 |
| 2 | 20000 | 0.007731 | 7347 | 0.002300 | 7332 |
| 3 | 30000 | 0.005475 | 11061 | 0.005903 | 11062 |
| 4 | 40000 | 0.010304 | 14717 | 0.006429 | 14664 |
| 5 | 50000 | 0.016473 | 18417 | 0.006652 | 18457 |
| 6 | 60000 | 0.017098 | 21971 | 0.009654 | 22096 |
| 7 | 70000 | 0.022270 | 25758 | 0.008805 | 25851 |
| 8 | 80000 | 0.016354 | 29571 | 0.009966 | 29687 |
| 9 | 90000 | 0.013908 | 33316 | 0.013783 | 33068 |
| 10 | 100000 | 0.013481 | 36865 | 0.011292 | 36779 |
| 11 | 110000 | 0.015708 | 40733 | 0.014511 | 40405 |
| 12 | 120000 | 0.015327 | 44138 | 0.013878 | 44337 |
| 13 | 130000 | 0.015719 | 47719 | 0.016542 | 48138 |
| 14 | 140000 | 0.015227 | 51647 | 0.016482 | 51647 |
| 15 | 150000 | 0.018250 | 55137 | 0.017276 | 55245 |
| 16 | 160000 | 0.018516 | 58898 | 0.018119 | 59024 |
| 17 | 170000 | 0.020040 | 62453 | 0.020960 | 62587 |
| 18 | 180000 | 0.020964 | 66184 | 0.021647 | 66063 |
| 19 | 190000 | 0.022675 | 69789 | 0.021845 | 69931 |
| 20 | 200000 | 0.025388 | 73534 | 0.023571 | 73571 |





**Контрольные вопросы**

**1) Что такое словарь, ассоциативный массив?**

**Словарь** (dictionary) — это структура данных для хранения пар вида

«ключ» — «значение» (key — value)

Альтернативное название — **ассоциативный массив** (associative array, map)

В словаре может быть только одна пара с заданным ключом

**2)Что такое бинарное дерево поиска? Проведите анализ сложности основных операций**

**Бинарное дерево** (binary tree) — это дерево (структура данных), в котором каждый узел (node)

имеет не более двух дочерних узлов (child nodes)

**Бинарное дерево поиска** (двоичное дерево поиска, binary search tree, BST) — это бинарное

дерево, в котором:

➔ Каждый узел x имеет не более двух дочерних узлов и содержит пару «ключ» — «значение»

➔ Ключи всех узлов левого поддерева x меньше значения его ключа

➔ Ключи всех узлов правого поддерева x больше значения его ключа

**Анализ эффективности BST**

Операции бинарного дерева поиска имеют трудоёмкость, пропорциональную высоте h дерева

В худшем случае высота дерева составляет O(n) —элементы добавляются в упорядоченной последовательности

В среднем случае высота дерева — O(logn)

**3)Что такое хеш-таблица? Проведите анализ сложности основных операций**

**Хеш-таблица** (hash table) — это структура данных для хранения пар «ключ» — «значение»

1. Доступ к элементам осуществляется по ключу (key)
2. Ключи могут быть строками, числами, указателями, ...
3. Хеш-таблицы позволяют в среднем за время O(1) выполнять добавление, поиск и удаление узлов



**4)Что такое хеш-функция? Какая хеш-функция является «хорошей»?**

**Хеш-функция** (hash function) — это функция, преобразующая значение ключа (например, строки, числа, файла) в целое число. Значение, возвращаемое хеш-функцией называется хеш-кодом (hash code), контрольной суммой (hash sum) или просто хешем (hash)

**Хеш-функция** преобразует (отображает) ключ (key) в номер элемента (index) массива — целое число от 0 до h – 1. Время вычисления хеш-функции зависит от длины ключа и не зависит от количества элементов в массиве Ячейки массива называются buckets, slots

**5)Методы разрешения коллизий в хеш-таблицах**

**Коллизия** (collision) — это совпадение значений хеш-функции для двух разных ключей

1. **Метод цепочек** (chaining) — закрытая адресация

Элементы с одинаковым значением хеш-функции объединяются в связный список. Указатель на список хранится в соответствующей ячейке хеш-таблицы

При коллизии элемент добавляется в начало списка

Поиск и удаление в худшем случае требуют просмотра всего списка

1. **Открытая адресация** (open addressing)

В каждой ячейке хеш-таблицы хранится не указатель на связный список, а один элемент (ключ, значение)

Если ячейка с индексом hash(key) занята, то осуществляется поиск свободной ячейки в следующих позициях таблицы