

## A2

Начнём с эксперимента, используя код на Python

Исходный код расположен в файле main.py (и в main.pdf, как удобнее).

- Переменная start - начальное размер хэш-таблиц
- Переменная end - конечный размер хэш-таблиц
- Переменная coeff - коэффициент от 0 до 1 обозначающий, насколько мы будем заполнять каждую таблицу. coeff умножается на размер таблицы, столько раз проводится insert

Программа циклом проходила с  $M = \text{start}$  до  $\text{end}$ , создавая хэш-таблицы размера  $M$  и делая insert  $M * \text{coeff}$  элементов. Т.е. coeff определяет, сколько часть таблицы мы заполняем. Если 1 – то всю, если 0.5 – то половину. В каждой итерации цикла вычисляется среднее число проб. После выполнения цикла, выполняется расчёт среднего количество проб для всего цикла.

В хэш-таблицу вносят случайные значения с помощью функции random.randint

Программа вычисляет среднее количество проб для квадратичного и кубического пробирования. Посмотрим на результаты экспериментов:

А. При коэффициенте заполненности, равном 1

```
Макс. размер хэш-таблиц = 100, коэф. заполнения = 1
Квадратичное пробирование:
Среднее число проб = 3.533012141211823
Кубическое пробирование:
Среднее число проб = 4.218802109767115
Сегодня квадратичное пробирование оказалось круче
```

```
Макс. размер хэш-таблиц = 250, коэф. заполнения = 1
Квадратичное пробирование:
Среднее число проб = 4.242952349732511
Кубическое пробирование:
Среднее число проб = 5.0180240034279615
Сегодня квадратичное пробирование оказалось круче
```

```
Макс. размер хэш-таблиц = 500, коэф. заполнения = 1
Квадратичное пробирование:
Среднее число проб = 4.746924581505348
Кубическое пробирование:
Среднее число проб = 5.790059622465767
Сегодня квадратичное пробирование оказалось круче
```

Макс. размер хэш-таблиц = 1024, коэф. заполнения = 1  
Квадратичное пробирование:  
Среднее число проб = 5.596853187953152  
Кубическое пробирование:  
Среднее число проб = 6.256527590723563  
Сегодня квадратичное пробирование оказалось круче

Макс. размер хэш-таблиц = 2500, коэф. заполнения = 1  
Квадратичное пробирование:  
Среднее число проб = 7.206787109375  
Кубическое пробирование:  
Среднее число проб = 7.9635726599290555  
Сегодня квадратичное пробирование оказалось круче

Макс. размер хэш-таблиц = 5000, коэф. заполнения = 1  
Квадратичное пробирование:  
Среднее число проб = 8.396186133852645  
Кубическое пробирование:  
Среднее число проб = 8.829789773613681  
Сегодня квадратичное пробирование оказалось круче

Можно заметить, что при необходимости заполнить хэш-таблицу максимально возможным количеством элементов, кубическое пробирование без шансов проигрывает квадратическому, ведь его число проб в среднем больше на +- 1

В. При коэффициенте заполненности, равном 0.5

Макс. размер хэш-таблиц = 100, коэф. заполнения = 0.5  
Квадратичное пробирование:  
Среднее число проб = 1.5747536689963877  
Кубическое пробирование:  
Среднее число проб = 1.591313230100625  
Сегодня квадратичное пробирование оказалось круче

Макс. размер хэш-таблиц = 250, коэф. заполнения = 0.5

Квадратичное пробирование:

Среднее число проб = 1.6611849272096595

Кубическое пробирование:

Среднее число проб = 1.63997190750335

Сегодня кубическое пробирование оказалось круче

Макс. размер хэш-таблиц = 500, коэф. заполнения = 0.5

Квадратичное пробирование:

Среднее число проб = 1.6641536669918702

Кубическое пробирование:

Среднее число проб = 1.660686163988001

Сегодня кубическое пробирование оказалось круче

Макс. размер хэш-таблиц = 1000, коэф. заполнения = 0.5

Квадратичное пробирование:

Среднее число проб = 1.6811740348362192

Кубическое пробирование:

Среднее число проб = 1.672043471961658

Сегодня кубическое пробирование оказалось круче

Макс. размер хэш-таблиц = 2000, коэф. заполнения = 0.5

Квадратичное пробирование:

Среднее число проб = 1.6835930587858041

Кубическое пробирование:

Среднее число проб = 1.673126495725277

Сегодня кубическое пробирование оказалось круче

Макс. размер хэш-таблиц = 5000, коэф. заполнения = 0.5

Квадратичное пробирование:

Среднее число проб = 1.683384610921159

Кубическое пробирование:

Среднее число проб = 1.678185003775311

Сегодня кубическое пробирование оказалось круче

Интересные вещи: при  $\text{coeff} = 0.5$  кубическое пробирование совсем немного, но выигрывает у квадратического. Хотя разница очень маленькая :)

Рассматривать коэффициенты ниже нет особо смысла, поскольку там вероятности коллизии будут сильно меньше.

Однако можно уже сделать выводы:

- Когда происходит заполнение хэш-таблицы более, чем на половину, квадратичное программирование показывает себя гораздо лучше с точки зрения возникновения кластеров и коллизий
- Когда хэш-таблица заполняется менее, чем на половину, кубическое программирование работает лучше

Кроме того, я не могу не упомянуть временами возникающие случаи, когда кубическое пробирование лучше квадратического. Вот примеры:

```
Макс. размер хэш-таблиц = 1000, коэф. заполнения = 1
Квадратичное пробирование:
Среднее число проб = 6.1368251873297
Кубическое пробирование:
Среднее число проб = 5.732365145228216
Сегодня кубическое пробирование оказалось круче
```

```
Макс. размер хэш-таблиц = 1000, коэф. заполнения = 1
Квадратичное пробирование:
Среднее число проб = 6.6328125
Кубическое пробирование:
Среднее число проб = 6.277777777777778
Сегодня кубическое пробирование оказалось круче
```

Однако эти случаи возникают только при «удачном» подборе параметров  $c_1$ ,  $c_2$  и  $c_3$ . При классической ситуации  $c_1 = c_2 = c_3 = 1$ , такого не наблюдается.

Для наглядности попробуем сделать параметры  $c_1$ ,  $c_2$  и  $c_3$  полностью случайными:

```
c1 = random.randint( a: 1, b: 100)
c2 = random.randint( a: 1, b: 100)
c3 = random.randint( a: 1, b: 100)
```

Однако даже в такой ситуации квадратичное пробирование окажется лучше:

```
Макс. размер хэш-таблиц = 1000, коэф. заполнения = 1
Квадратичное пробирование:
Среднее число проб = 6.885429877628053
Кубическое пробирование:
Среднее число проб = 7.343713346183137
Сегодня квадратичное пробирование оказалось круче
```

Почему так происходит?

Оба метода основаны на идее изменения хеш-значения при коллизии, чтобы избежать кластеризации. Однако, очевидно, что у кубического пробирования гораздо более быстрый экспоненциальный рост с ростом числа попыток  $i$

В результате, на практике кубическое пробирование увеличивает вероятность образования кластеров и менее равномерно распределяет значения, что в итоге приводит к худшей производительности в случае большой заполненности хэш-таблицы.

Квадратичное пробирование, с другой стороны, может обеспечивать более равномерное распределение значений при увеличении номера попытки  $i$ , что способствует уменьшению вероятности коллизий.

Однако при малой заполненности хэш-таблицы быстрый экспоненциальный рост кубического пробирования играет только на руку: быстрое перемещение по ячейкам хэш-таблицы даёт небольшое преимущество. Но этой разницы недостаточно, чтобы компенсировать сложность вычисления  $i * i * i$

Таким образом, квадратичное пробирование работает эффективней, чем кубическое, как с точки зрения образования кластеров и возникновения коллизий, так и сложности вычислений.