

АНАЛИЗ ЭФФЕКТИВНОСТИ РАБОТЫ АЛГОРИТМОВ ПОИСКА

Поиск всегда был и будет необходим тем, кто использует информацию в любом виде: как цифровом, так и физическом.

Цель работы заключается в описании наиболее распространенных алгоритмов поиска, анализе эффективности алгоритмов поиска информации в малых (30 записей), средних (1000 записей) и больших (100 тыс. записей) объемах данных различными алгоритмами.

Для анализа сложности алгоритмов вводят понятие «О большого». О-нотация или Big O, используют для обозначения максимального времени в операциях, которое может понадобиться алгоритму.

Так, если для некоторого алгоритма указано $O(\log n)$, это значит, что максимальное время прохождения этого алгоритма на 16 записях будет занимать по времени 4 операции.

В работе представлены алгоритмы линейного поиска, которому соответствует $O(n)$, и бинарного, которому соответствует $O(\log n)$.

Линейный поиск основан на принципе «начать с начала и продолжать, пока не будет найден искомый ключ; затем остановиться».

Этот метод поиска – один из наиболее популярных вследствие своей простоты в реализации и понятности. Алгоритм линейного поиска представлен в виде псевдокода:

```

пока не прошли последний элемент
{
  если элемент последовательности входит в диапазон значений
  {
    запомнить элемент;
    выйти из цикла
  }
}

```

Бинарный поиск сравнивает средний элемент диапазона с искомым значением и в зависимости от результата сравнения поиск либо повторяется на одной из половин таблицы, либо выводится полученный результат.

Алгоритм имеет большую скорость выполнения, но его сложнее реализовать. Бинарные алгоритмы требуют отсортированных данных, что делает их не всегда возможными к применению.

Псевдокод бинарного поиска:

```

пока позиция начала интервала меньше позиции конца интервала
{
    определить середину интервала;
    если элемент в середине больше искомого значения,
    то сместить конец интервала в его середину,
    иначе сместить начало на следующий после середины элемент;
}
уменьшить конец на 1;
если элемент в итоговом конце входит в диапазон, то запомнить элемент;

```

Табличный метод заключается в выделении на основе общей таблицы с данными производных таблиц с ключевыми полями, по которым можно обращаться к необходимым записям. Одно из полей записи можно использовать в качестве индекса, а остальные данные для этого поля можно записывать в отдельную таблицу. Тогда алгоритм можно свести к единственной проверке: существует ли запись с таким полем, то есть алгоритму соответствует $O(1)$. При поиске значений в некотором заданном диапазоне алгоритм выполняется быстрее, т.к. уже не приходится тянуть за собой огромные записи, а только лишь отдельные их поля.

По результатам проведенной работы можно констатировать тот факт, что при малых и средних объемах исходных данных однозначного фаворита нет, но на больших объемах данных мы видим, что линейный алгоритм требует существенно больше времени, чем остальные.

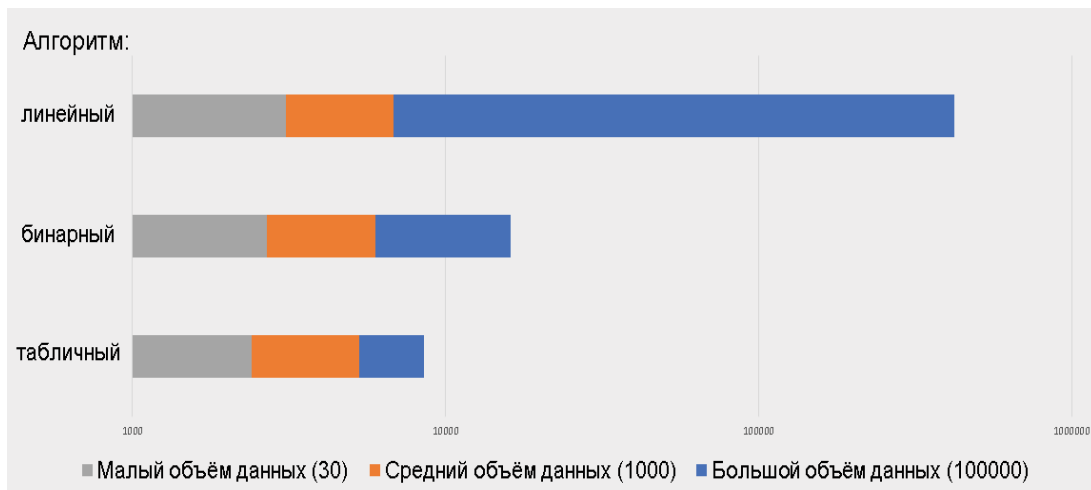


Рисунок 1 – Сравнение алгоритмов

Для решения задачи сравнения эффективности работы алгоритмов поиска на любых случайно генерируемых диапазонах реализована программа и получен следующий результат: при расширении диапазона для поиска линейный алгоритм по скорости поиска опережает

бинарный поиск. А табличный метод наиболее эффективен при любом диапазоне выборки.

Практическое применение данных алгоритмов я осуществил на примере выборки студентов по среднему баллу. Мною было решено применить только линейный и табличный методы как более интересные для анализа результатов, исключив бинарный по причине его крайней неэффективности при решении этой задачи. Как и предполагалось, поиск по табличному алгоритму оказался более быстрым и эффективным.

ЛИТЕРАТУРА

1. Кнут Д. Э. Искусство программирования. Том 3. Сортировка и поиск / под ред. В. Т. Тертышного (гл. 5) и И. В. Красикова (гл. 6). — 2-е изд. — Москва: Вильямс, 2007. — Т. 3. — 832 с.

2. Макконнелл С. Совершенный код. Мастер-класс / Пер. с англ. — издательство «Русская редакция», 2010. — 896 стр.

3. CodeLessons Бинарный поиск по массиву C++ [Электронный ресурс]. — Режим доступа: <https://codelessons.ru/cplusplus/algoritmy/binarnyj-poisk-po-massivu-c.html>. — Дата доступа: 15.03.2021

УДК 004.415.2

Студ. Е.А. Гончар

Науч. рук. зав. кафедрой Н.В. Пацей
(кафедра программной инженерии, БГТУ)

СИСТЕМА ОТСЛЕЖИВАНИЯ ОШИБОК И УПРАВЛЕНИЕ ПРОЕКТОМ

Система отслеживания ошибок (англ. *bug tracking system*) – прикладная программа, разработанная с целью помочь разработчикам программного обеспечения (программистам, тестировщикам и др.) учитывать и контролировать ошибки (баги), найденные в программах, а также следить за процессом устранения этих ошибок.

Целью работы была разработка веб-приложения «*Issue Tracking System*». Приложение содержит две части: серверную, разработанную на языке *Java* и фреймворка *Spring*, и клиентскую, в которой использовался *Java Script* с фреймворком *Angular*.

Функционально приложение содержит две роли пользователь и администратор. Пользователи без прав администратора могут создавать, редактировать, удалять и фильтровать задачи и ошибки, а также управлять профилем. Администратор может выполнять все вышеперечисленные функции, а также создавать новых пользователей, редактировать информацию о них и удалять.