

УДК 004.023 [519.178+004.043] 004.424 615.035.4 621.372 681.5.14

Бинарный поиск

Пасько Д.П.

Научный руководитель – к.т.н., доцент СИЗИКОВ С.В.

Ежедневно все люди сталкиваются с поиском. Для каждой цели существует свой алгоритм поиска и условно их можно поделить на поиск в неупорядоченном и поиск в упорядоченном множестве данных.

Упорядоченным или отсортированным множеством в общем виде можно назвать множество, в котором элементы «расставлены» последовательно по какому-либо критерию (например: по возрастанию/убыванию), что облегчает дальнейшую работу с этими элементами.

Двоичный (бинарный) поиск — один из классических, наряду с линейным алгоритмом поиска.

Впервые был опубликован в 1946 году Дональдом Кнутом, создателем знаменитого труда «Искусство программирования».

В некоторых случаях данный тип поиска может быть эффективней линейного поиска.

Суть поиска заключается в том, что проверяется искомое значение на то, больше ли оно, меньше или равно значению в середине массива.

Можем разобрать по шагам **ход работы алгоритма**:

1. Выбираем зону поиска.
2. Средний элемент сравнивается с искомым.
3. Если для проверки не осталось элементов, то алгоритм завершается, иначе выполняется пункт 1.

Хочу напомнить, что хоть этот метод и кажется идеальным, но не стоит забывать, что применим он только к отсортированному массиву, в то время как, например, линейный поиск, будет проверять значения один за другим вне зависимости от сортировки.

Мы рассмотрели не только простой, но и теоретически идеальный случай, однако в реальной жизни в данном, казалось бы, простом методе может возникнуть куча ошибок, например:

- не работает с массивом их 0/1/2 элементов;
- не находит первый или последний элемент;
- некорректно работает, если элемента в массиве нет;
- некорректно работает, если в массиве есть повторяющиеся элементы.

И под конец хотелось бы заметить, что если данных немного (как в нашем примере), то значительной разницы в скорости работы двух методов (бинарный и линейный) не будет. Однако если массив будет достигать 10-ти тысяч элементов, то двоичный поиск будет работать на 72% быстрее, чем обычный цикл, то есть почти в 2 раза!

Литература

1. Кормен, Т. Алгоритмы: построение и анализ / Т. Кормен [и др.]; пер. с англ. – М.: Вильямс, 2007. – 1296 с.
2. Седжвик, Р. Алгоритмы на C++ / Р. Седжвик; пер. с англ. – М.: Вильямс, 2011. – 1056 с.
3. Шилдт Г. C++: Руководство для начинающих.