

Специальная нотация «О-большое» описывает скорость работы алгоритма. Зачем вам это? Время от времени вам придется использовать чужие алгоритмы, а потому неплохо было бы понимать, насколько быстро или медленно они работают. В этом разделе я объясню, что представляет собой «О-большое», и приведу список самых распространенных вариантов времени выполнения для некоторых алгоритмов.



Боб пишет алгоритм поиска для NASA. Его алгоритм заработает, когда ракета будет подлетать к Луне, и поможет вычислить точку посадки.

Это один из примеров того, как время выполнения двух алгоритмов растет с разной скоростью. Боб пытается выбрать между простым и бинарным поиском. Его алгоритм должен работать быстро и правильно. С одной стороны, бинарный поиск работает быстрее. У Боба есть всего 10 секунд, чтобы выбрать место посадки; если он не уложится в это время, то момент для посадки будет упущен. С другой стороны, простой поиск пишется проще и вероятность ошибок в нем ниже... Конечно, Боб *совершенно* не хочет допустить ошибку в коде посадки ракеты. И тогда для пущей уверенности Боб решает измерить время выполнения обоих алгоритмов для списка из 100 элементов.

Допустим, проверка одного элемента занимает 1 миллисекунду (мс). При простом поиске Бобу придется проверить 100 элементов, поэтому поиск займет 100 мс. С другой стороны, при бинарном поиске достаточно проверить всего 7 элементов ($\log_2 100$ равен приблизительно 7), а поиск займет 7 мс. Но реальный список может содержать более миллиарда элементов. Сколько времени в таком случае потребуется для выполнения простого поиска? А при бинарном поиске? Обязательно ответьте на оба вопроса, прежде чем продолжить чтение.