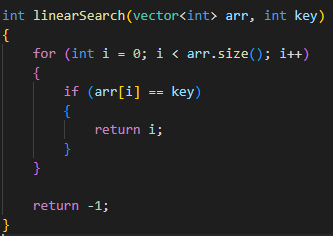
**Тема**: Алгоритмы поиска

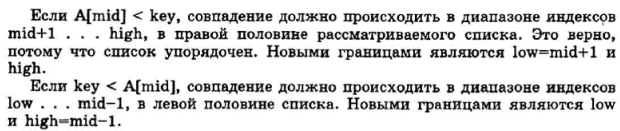
**Источники**: «Структуры данных в C++» - У. Форд, У. Топп. 1999 г.

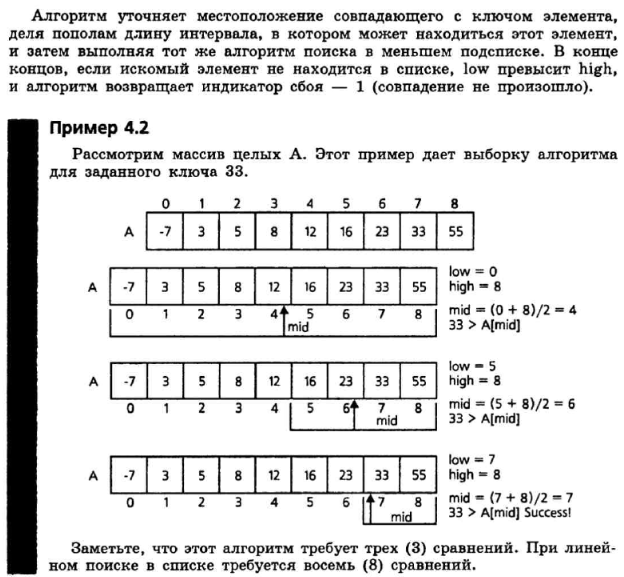
Алгоритмы поиска

Существует два основных алгоритма поиска: последовательный (линейный) и бинарный.  
 **Линейный** алгоритм поиска является последовательным прохождением по всему массиву, до того момента, пока не найдётся искомый элемент (ключ). В наилучшем варианте развития событий первый же элемент будет искомым – O(1). В наихудшем варианте ключ не будет найден или он окажется в самом конце поиска – O(n). Усреднённый вариант составляет половину от n – O(n / 2). Ниже приведён пример реализации линейного поиска в одномерном массиве ключа на языке C++.



**Бинарный** алгоритм кардинально отличается от линейного, но и применим не ко всем массивам (в отличии от линейного), только к отсортированным. Так как массив отсортирован, мы понимаем, что наименьший по значению элемент находиться слева, а наибольший – справа (Или, наоборот, если массив отсортирован по убыванию). Потому мы должны разделять массив на отрезки, а дальше сравнивать число с ключом: если ключ меньше, то идём анализировать левый отрезок, если больше – правый. Или сразу находим ключ. А затем мы повторяем обработку отрезка, пока не найдём искомое, или пока не поймём, что ключа в массиве нет.



Бинарный поиск работает куда быстрее линейного:

* Худший вариант – O(log(2) n)
* Наилучший случай (когда ключ находиться в середине) – O(1)
* Средний вариант - O((log(2) n) / 2)

Но, стоит повторить, бинарный поиск работает только для отсортированных массивов. Ниже предоставлена реализация функции бинарного поиска по ключу на языке C++.

