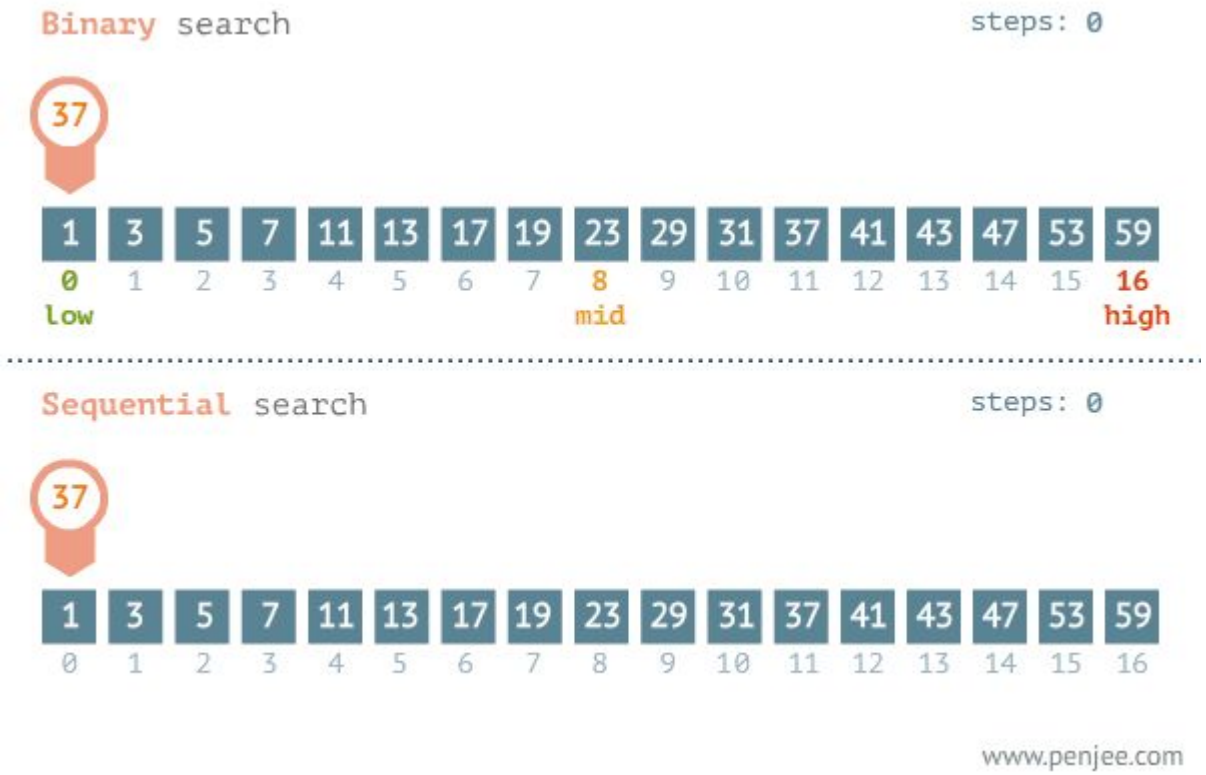
Бинарный поиск

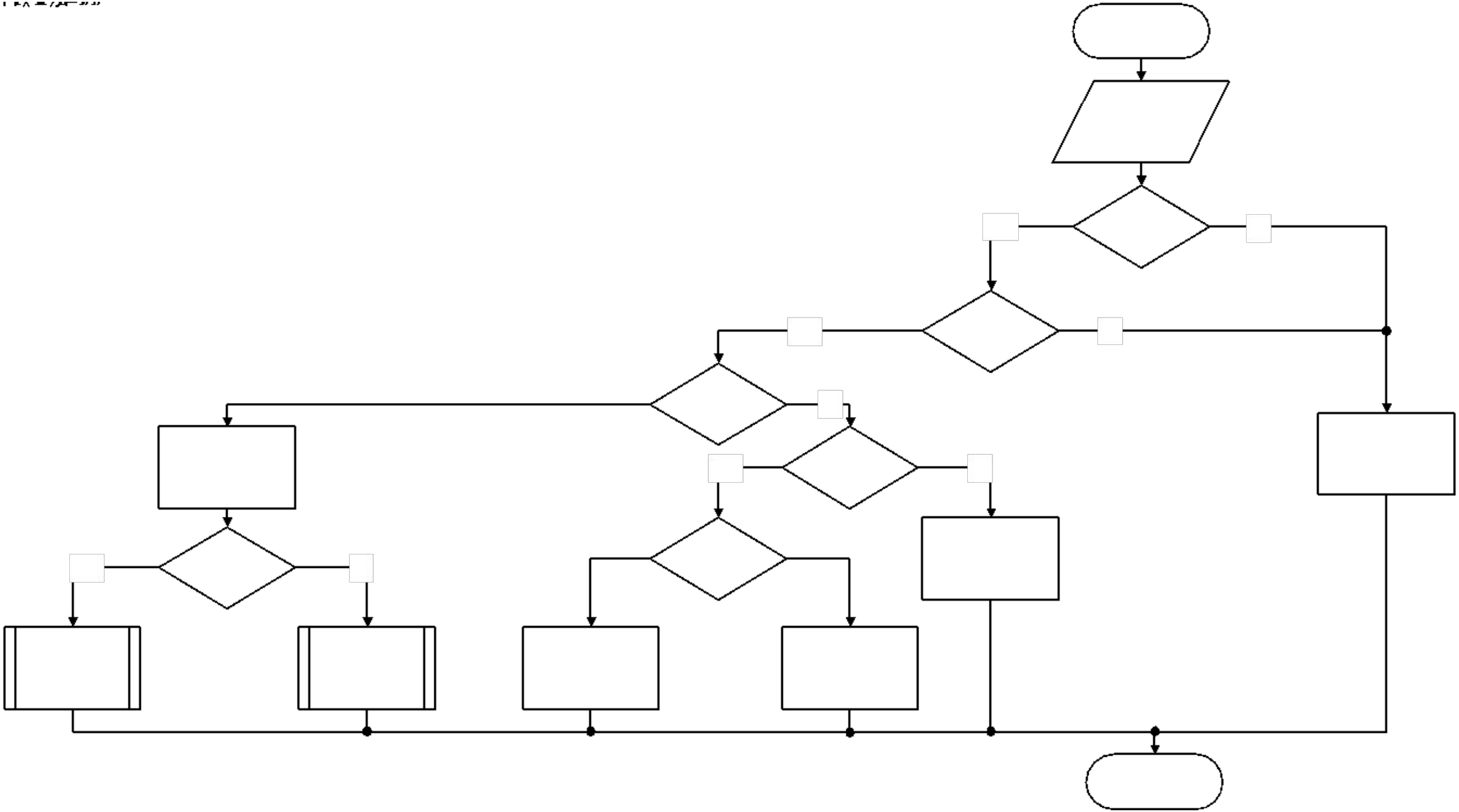
ОПРЕДЕЛЕНИЕ

Бинарный поиск – алгоритм поиска элемента в отсортированном массиве. Основа метода – деление массива (области поиска) на половины.



Алгоритм

1) Проверить, что искомое значение не выходит за границы области поиска;
2) Найти середину области поиска;
3) Если искомое значение больше, чем значение в середине области, то повторяем поиск в области от середины до наибольшего значения области, иначе – от наименьшего до середины.



ПРИМЕР РЕАЛИЗАЦИИ

```
int binary_search (int value, int *array, int *first, int *last){
    if ((value>*(last))||(value<*first))
        return -1;
    else if (last-first==1)
    {
        if (*first==value) return array-first;
        else if (*(last)==value) return last-array;
        else return -1;
    }
    else {
        unsigned int mid = (last - first) / 2;
        if (value > *(first + mid))
            return binary_search(value, array, first + mid, last);
        else
            return binary_search(value, array, first, last - mid);
    }
}
```

ТИПИЧНЫЕ ОШИБКИ

1) first+last вызывает выход за границы диапазона используемого типа данных. Решается использованием указателей или итераторов.
2) Ошибки на единицу.
3) Ищется не первое/последнее значение.

STL

bool binary_search – возвращает **истину**, если искомый элемент встречается в массиве и **ложь** в противном случае.

```
binary_search(array_100,array_100+100,63);
```

СКОРОСТЬ РАБОТЫ

Число итераций для поиска некоторого числа в массиве	
Бинарный поиск	Линейный поиск
7	57
10	500
14	6005
17	20025

Сложность линейного поиска – $O(n)$
Сложность бинарного поиска – $O(log_2 n)$

Замечания

Значение mid – не обязательно середина массива. Оно определяется как граница раздела двух областей поиска – той, в которой будет продолжен поиск, и той, которая будет отброшена.

Большинство задач на бинарный поиск подразумевают именно правильное определение mid.

Единого алгоритма для данного алгоритма не существует. В каждом случае mid определяется только на основе анализа задачи.

Замечания

Бинарный поиск работает с любыми структурами данных, которые расположены в памяти последовательно и **отсортированы**.

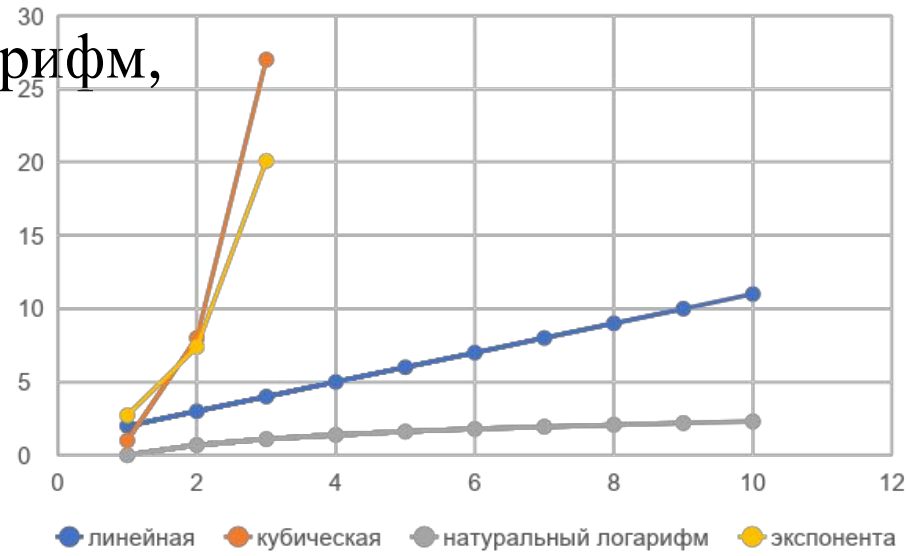
В случае использования таких структур данных, как стек, очередь, дерево и т.д. бинарный поиск не работает.

Замечания

Бинарный поиск можно использовать для монотонных функций.

Функция монотонна в том случае, если она всё время не убывает или не возрастает.

Примеры – линейная, кубическая, логарифм, экспонента.



Литература

- Кнут, «Искусство программирования», том 3;
- Лафоре, «Структуры данных и алгоритмы Java» (тут хватит и знаний C++).
- Вирт Н. Алгоритмы + структуры данных = программы.