Алгоритмы поиска

Поиск – процесс нахождения конкретной информации в ранее созданном множестве данных.
Одно из наиболее часто встречаемых действий в программировании.

- Для поиска важно выбрать алгоритм, который лучше всего подходит для конкретной задачи.
- Основные идеи алгоритмов поиска сосредоточены на методах перебора и стратегии поиска.
- Алгоритмы поиска делятся на линейные и бинарные.

1. Вычисление элемента, что часто предполагает получение значения элемента.
2. Сравнение элемента с эталоном.
3. Перебор элементов множества.

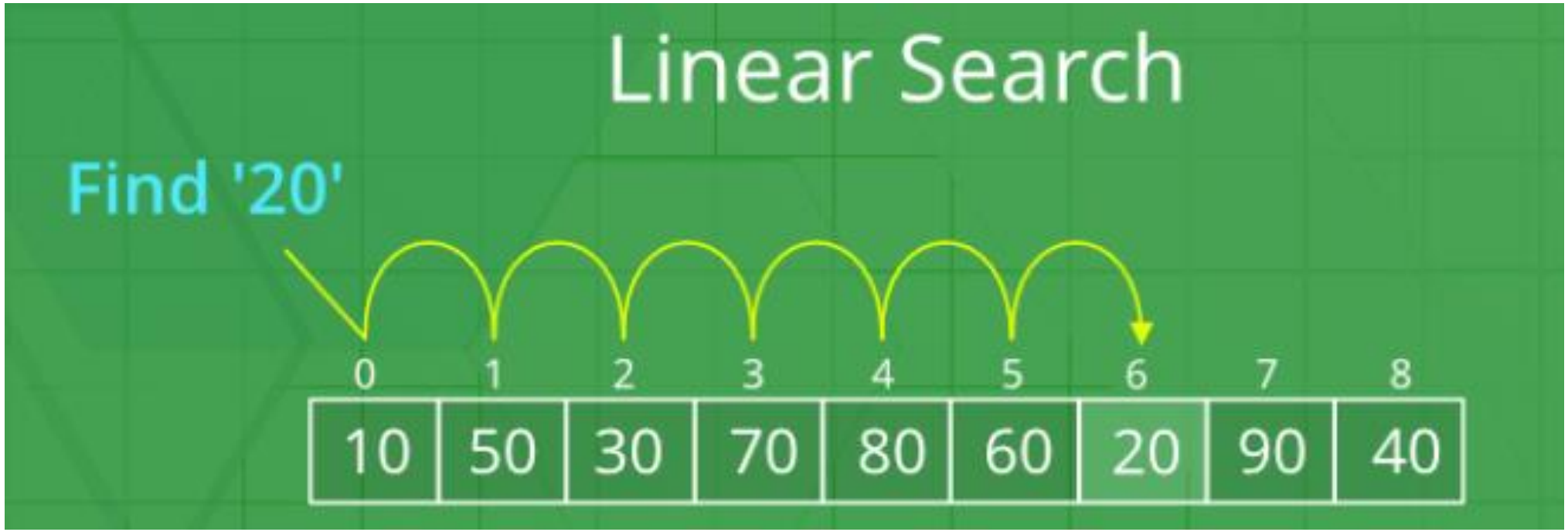
Linear Search

Простейший вид поиска элемента в множестве путём последовательного сравнения очередного рассматриваемого значения с искомым до тех пор, пока эти значения не совпадут.

1. Переход к следующему элементу.
2. Если текущий элемент равен искомому – поиск прекращается.
3. В противном случае выполняется п1.
4. В конце если элемент не найден - возвращается ошибка.

Сложность поиска в худшем случае O(N)

Linear Search



Linear Search

```
void linear_search(vector<int> &v, int target, int *result) {
    for (int i = 0; i < v.size(); ++i) {
        if (target == v[i]) {
            *result = i;
            return;
        }
    }

    result = nullptr;
}
```

А можно проще?

godbolt.org

Linear Search

```
void linear_search(vector<int> &v, int target, int *result) {
    for (int i = 0; i < v.size(); ++i) {
        if (target == v[i]) {
            *result = i;
            return;
        }
    }

    result = nullptr;
}

void linear_search(vector<int> &v, int target,
int * result) {
    auto it = find(v.begin(), v.end(), target);

    *result = it - v.begin();
}
```

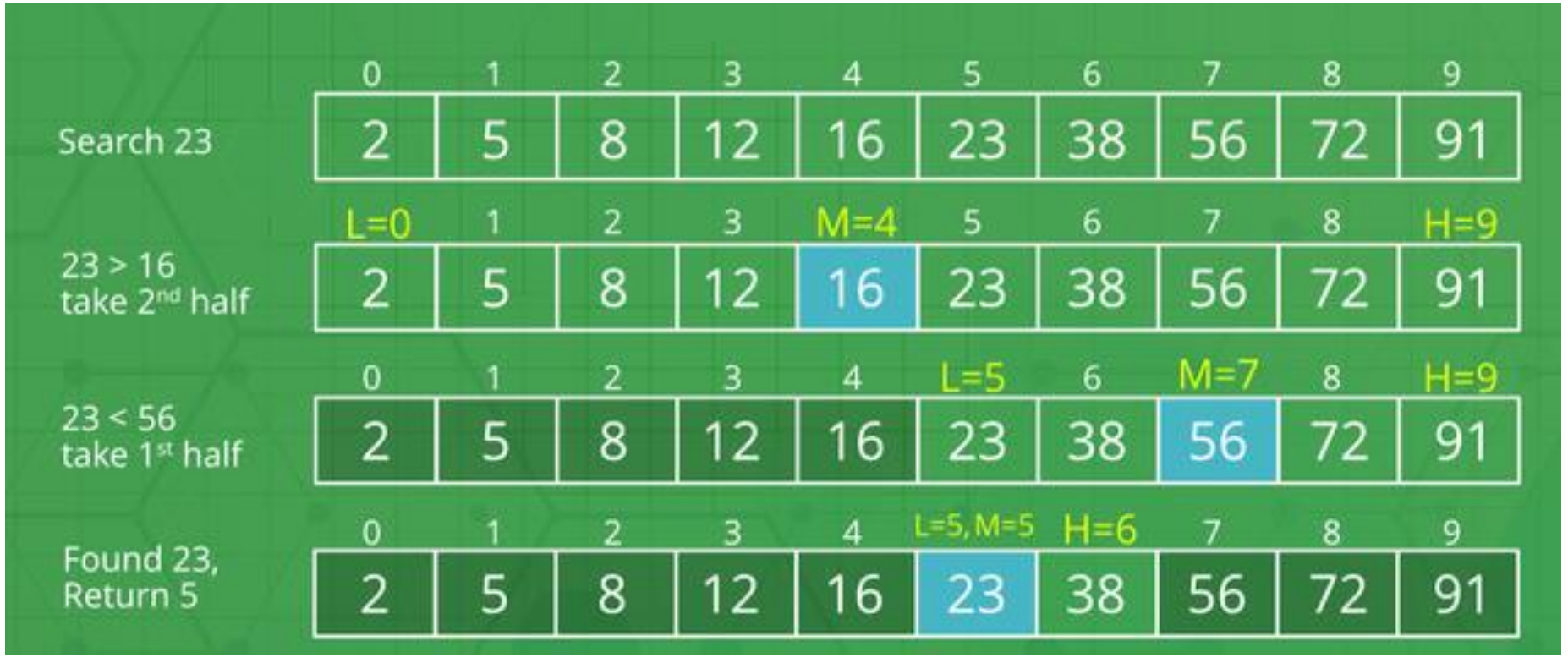
godbolt.org

Binary Search

1. Определение значения центрального (опорного) элемента множества.
2. Значение опорного элемента сравнивается с искомым значением.
3. В зависимости от результатов сравнения выделяется подмножество слева или справа от опорного элемента, в котором будет продолжаться поиск с п.1
4. Поиск продолжается пока опорный элемент не равен искомому, или из множества нельзя выделить подмножество.

- Сложность такого поиска O(log N), что существенно быстрее.
- Но какое обязательное условие для такого поиска?

Binary Search



Binary Search

```
int binary_search(vector<int>& v, int x) {
    int begin = 0; int end = v.size() - 1;

    while (begin <= end) {
        int mid = (begin + end) / 2;

        /* x founded */
        if (v[mid] == x)  return mid;

        /* x at the left side */
        else if (v[mid] > x) end = mid - 1;

        /* x at the right side */
        else begin = mid + 1;
    }

    return -1;
}
```

godbolt.org