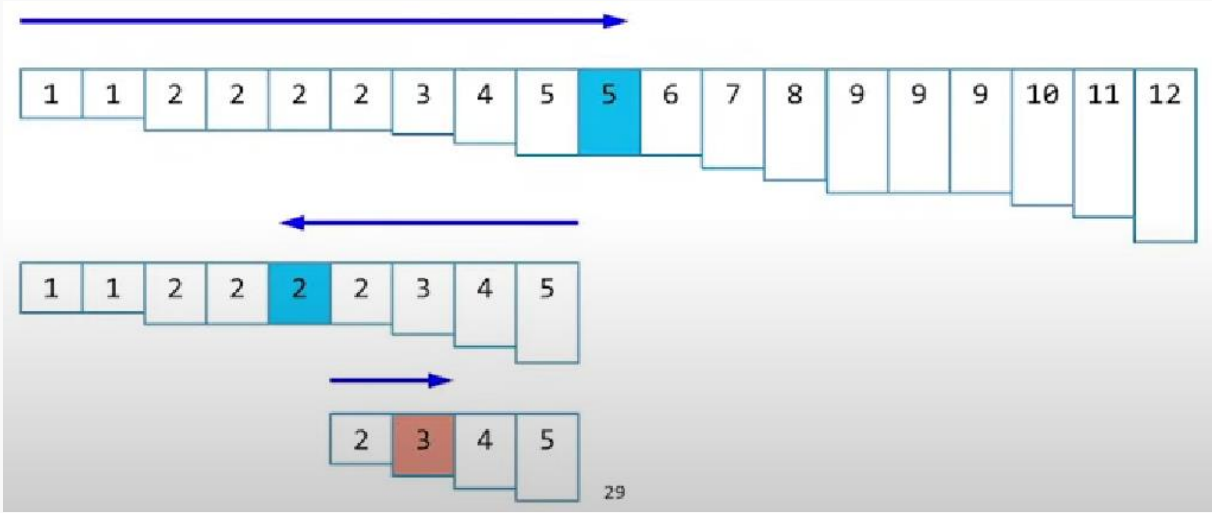
ЛЕКЦИЯ 11
РАЗДЕЛЯЙ И ВЛАСТВУЙ

АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ

ЛЕКТОР ФУРМАВНИН С.А.

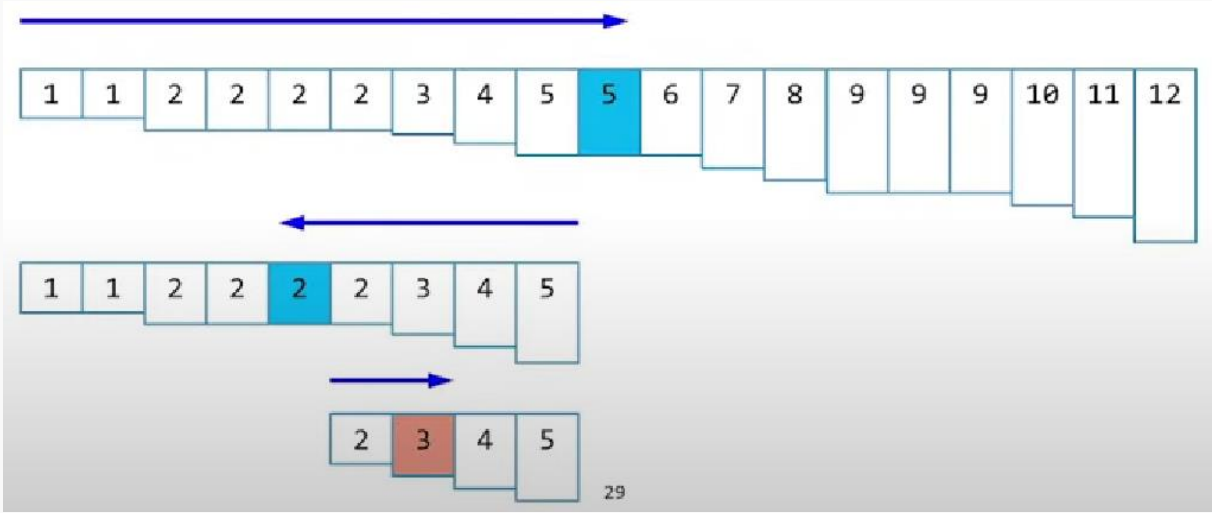
СТРАТЕГИЯ РАЗБИЕНИЯ ПОПОЛАМ

Также известна как «Divide and Conquer»



СТРАТЕГИЯ РАЗБИЕНИЯ ПОПОЛАМ

Также известна как «Divide and Conquer»



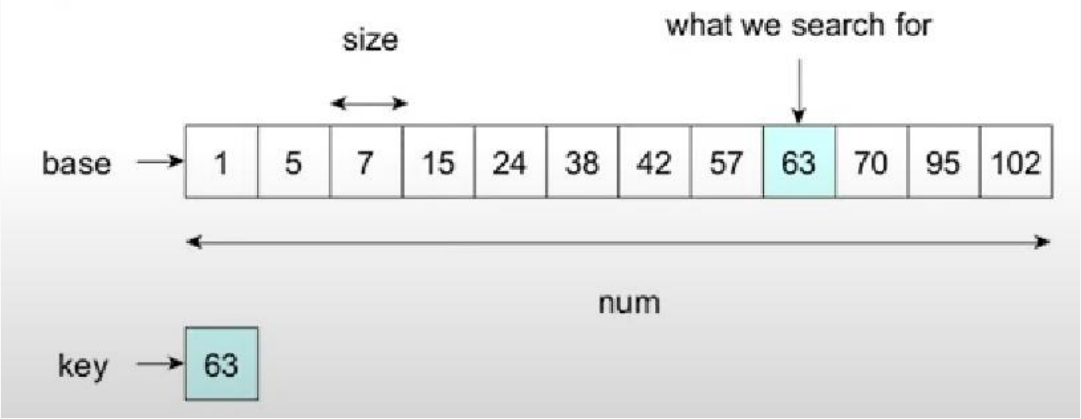
АЛГОРИТМ БИНАРНОГО ПОИСКА

```
unsigned binary_search(int const * parr, unsigned len,
int elem){
	int l = 0; int r = len - 1;
	while (l <= r) {
		int m = l + (r - l) / 2;
		if (parr[m] == elem) return m;
		if (parr[m] > elem) l = m - 1;
		if (parr[m] < elem) l = m + 1;
	}
	return len;
}
```

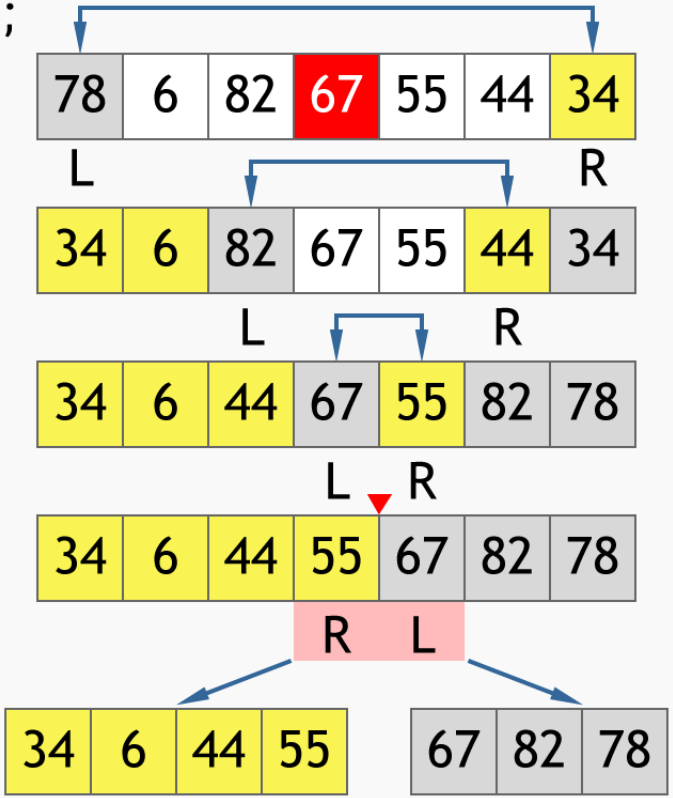
АЛГОРИТМ ОБЩЕГО БИНАРНОГО ПОИСКА

```
typedef int (*cmp_t)(const int& lhs, const int& rhs);

unsigned binary_search(int const * parr, unsigned len,
int elem, cmp_t cmp);
```



БЫСТРАЯ СОРТИРОВКА

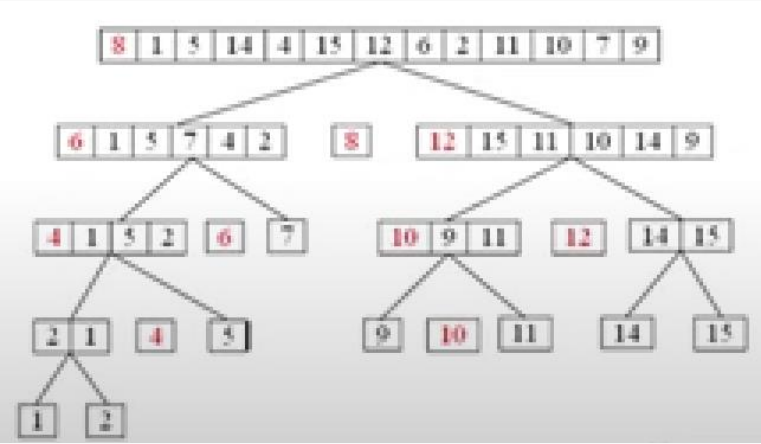


- Массив делится на две части по pivot (можно всегда выбирать первый)
- Далее результаты разбиения сортируются отдельно этим же способом
- В итоге массив собирается из отсортированных подмассивов

БЫСТРАЯ СОРТИРОВКА

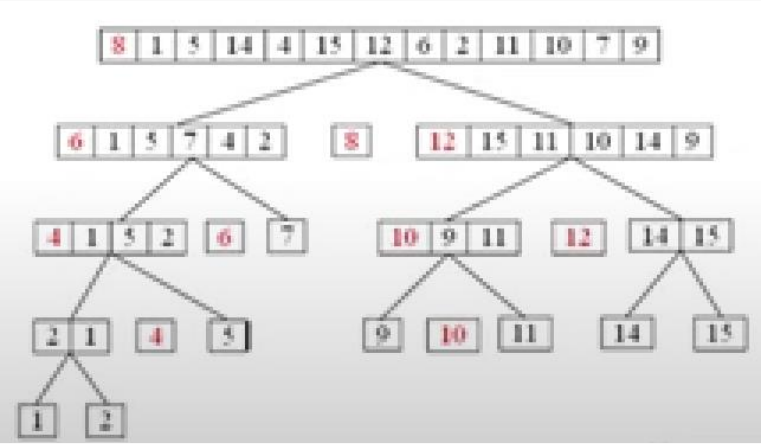
Ниже представлен средний случай

Как выглядит худший случай?

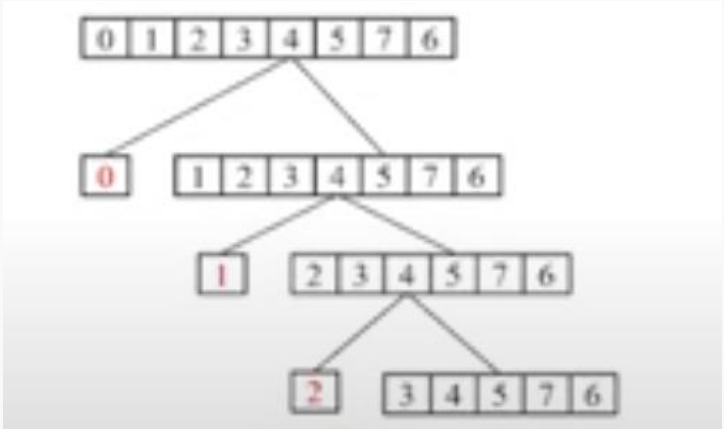


БЫСТРАЯ СОРТИРОВКА

Ниже представлен средний случай



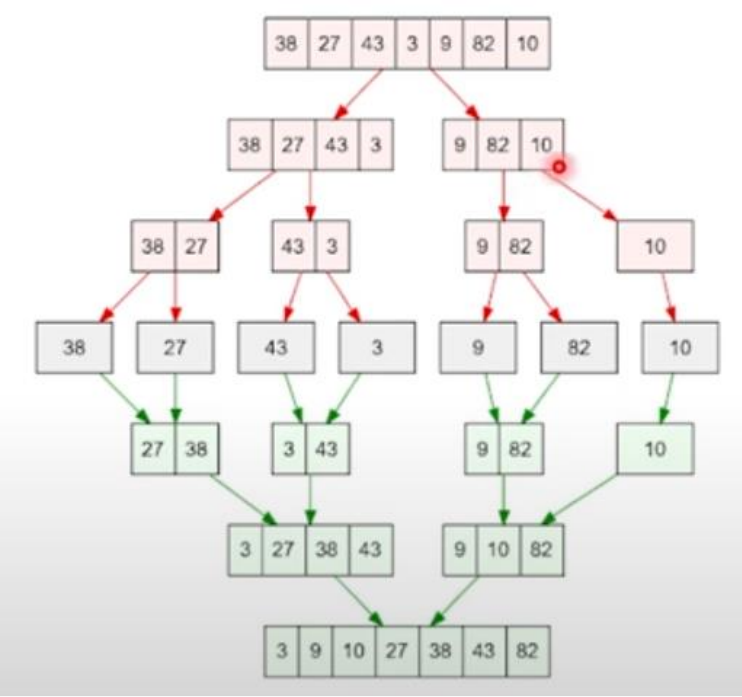
Как выглядит худший случай?



ОБСУЖДЕНИЕ

- Быстрая сортировка в худшем случае все ещё ведет себя как $O(n^2)$
- Есть хитроумные способы избежать на входе почти отсортированных массивов: например случайно перемешивать массив перед сортировкой
- Можно ли придумать сортировку, которая **всегда** работает за $O(n \log n)$?

СОРТИРОВКА СЛИЯНИЕМ



- Делим массив на каждом шаге примерно пополам
- Во все не обязательно при этом реально выделять новые массивы, можно хранить старые индексы
- Далее сливаем получившиеся подмассивы и получаем отсортированные подмассивы
- У нас нет худшего случая

СОРТИРОВКА СЛИЯНИЕМ

Вам предлагается реализовать ключевой шаг: слияние

```
void merge(int *arr, int l, int m, int r) {
      // TODO: Напишите ваш код здесь}

void merge_sort(int *arr, int l, int r) {
      if (l >= r) return;
      int m = (l + r) / 2;
      merge_sort(arr, l, m);
      merge_sort(arr, m + 1, r);
      merge(arr, l, m, r);
}
```

ПЕРЕМНОЖЕНИЕ ПОЛИНОМОВ

- Пусть даны $A(x) = x^3 + 3x^2 + 4x + 7$ и $B(x) = x^3 + 5x^2 + x + 4$
- Чему равно $A(x) \cdot B(x)$?
- Постарайтесь осознать КАК вы это посчитали?

КОРОТКО О СТРУКТУРАХ

- В языке C и C++ есть возможность создавать свои типы данных
- Как правило, часто требуется объединить несколько переменных разных типов в единую конструкцию – структуру.
- Синтаксис структур довольно прост:

```
struct Point {
   int x;
   int y;
};
```

КОРОТКО О СТРУКТУРАХ

- Синтаксис структур довольно прост:

```
struct Point {
   int x;
   int y;
};
```

  - Далее в коде мы можем создать переменную своего типа:

```
Point A{3, 5};
```

  - Для обращения к полям структуры используется точка или `->`, если это указатель на структуру.

КОРОТКО О СТРУКТУРАХ

- Далее в коде мы можем создать переменную своего типа:

```
Point A{3, 5};
```

- Для обращения к полям структуры используется точка или ->, если это указатель на структуру.

```
A.x = 2; A.y = 3; // Теперь это A(2; 3)
Point* pA = &A; // pA указывает на A
pA.x = 1; // CE
pA->x = 1; // OK, теперь это A(1; 3)
(*pA).x = 1; // Тоже, что и pA->x = 1
Point& refA = A; // Тоже самое, что A
refA.x = 4; // OK, теперь это A(4; 3)
refA->x = 2; // CE
```

ПЕРЕМНОЖЕНИЕ ПОЛИНОМОВ

Пусть даны $A(x) = a_0x^n + \cdots + a_n$ и $B(x) = b_0x^m + \cdots + b_m$
Вам необходимо посчитать их произведение самым простым и очевидным способом: последовательно перемножая коэффициенты

```
struct Poly{ unsigned n; unsigned* p; };

Poly mult(const Poly& lhs, const Poly& rhs) {
      Poly ret = {rhs.n + lhs.n - 1, nullptr};
      ret.p = new unsigned[ret.n];
      // TODO: Ваш код здесь
      return ret;
}
```

ДОМАШНЕЕ ЗАДАНИЕ

1. Дан целочисленный массив, содержащий не менее четырех элементов. Напишите функцию, которая принимает указатель на этот массив, его размер и возвращает сумму двух самых маленьких по модулю чисел в нём.
2. Дан целочисленный массив, содержащий не менее четырех элементов. Напишите функцию, которая принимает указатель на этот массив, его размер и возвращает разность между самым большим и самым маленьким значениями в нем.
3. Дан целочисленный массив, содержащий не менее четырех элементов. Напишите функцию, которая принимает указатель на этот массив, его размер и индекс элемента и возвращает индекс самого маленького элемента, который больше заданного. Если такого нет, то верните -1.

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1. Дорохова Т.Ю., Основы алгоритмизации и программирования : учебное пособие для СПО / Т.Ю. Дорохова, И.Е. Ильина. – Саратов, Москва : Профобразование, Ай, Пи Ар Медиа, 2022. – 139 с.
2. Кудинов Ю.И., Основы алгоритмизации и программирования : учебное пособие для СПО / Ю.И. Кудинов, А.Ю. Келина. – 2-е изд. – Липецк, Саратов: Липецкий государственный технический университет, Профообразование, 2020. – 71 с.
3. Дональд Кнут, Искусство программирования. Том 1. Основные алгоритмы / Ю.В. Козаченко. - 3-е изд – Москва, Санкт-Петербург: ВИЛЬЯМС, 2018. – 721 с.