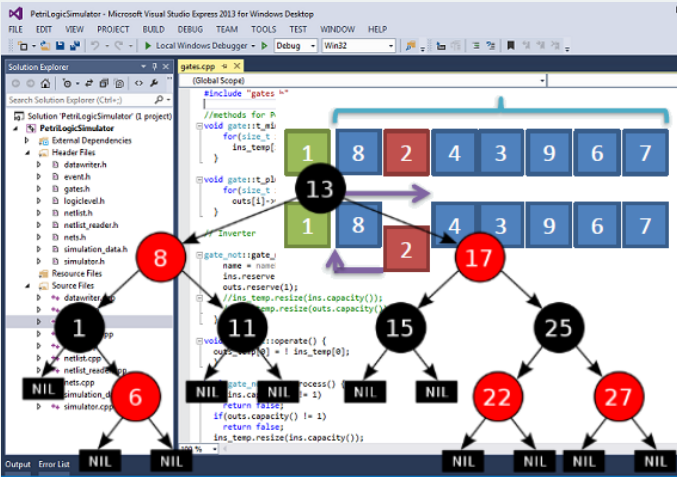




Теория алгоритмов

Лекция 3

Обработка линейных структур данных.
Алгоритмы сортировки и поиска.



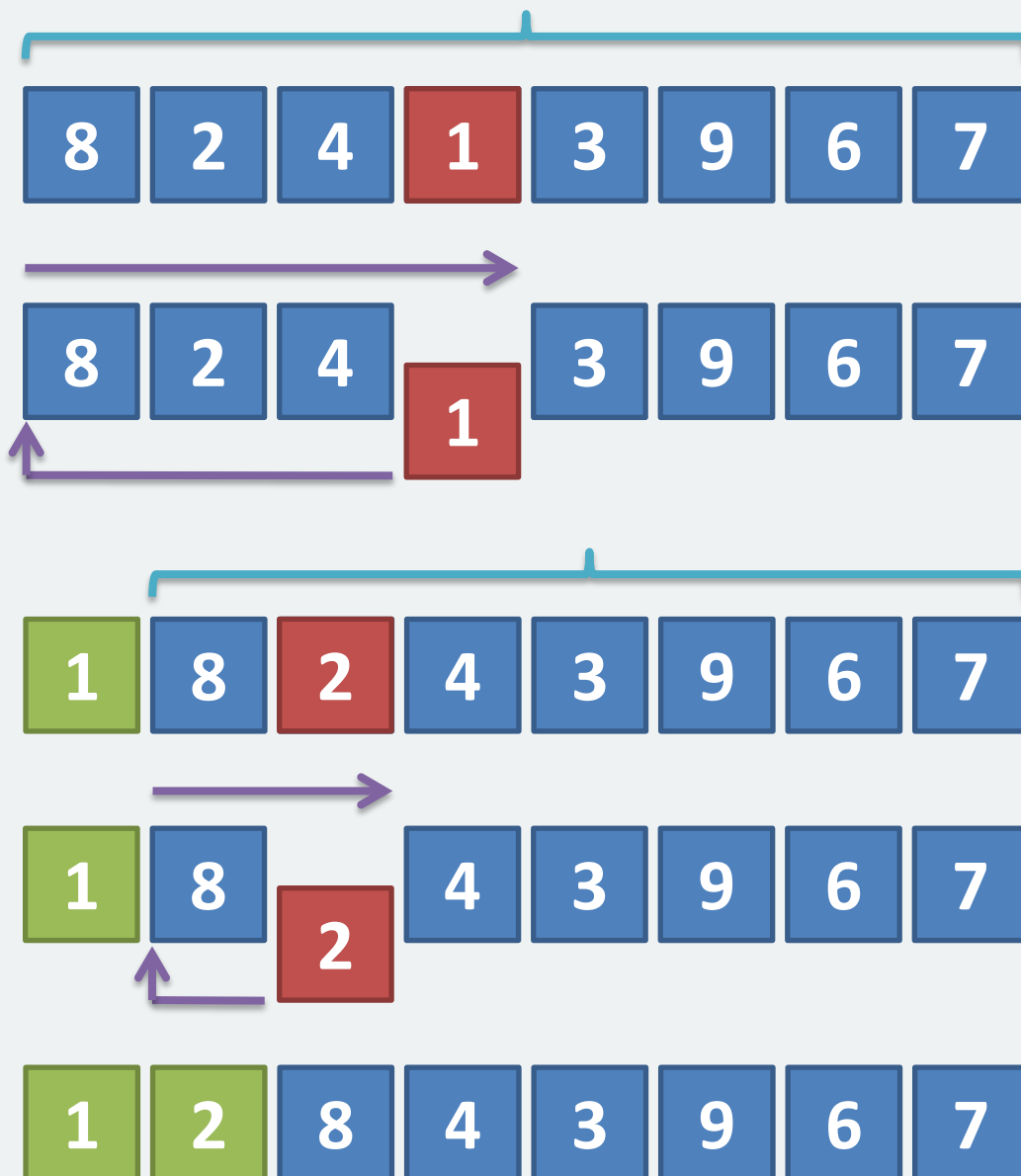
Характеристики алгоритмов сортировки

1. Отношение к расположению данных:
 - во внутренней памяти;
 - во внешней памяти.
2. Вычислительная сложность:
 - квадратичная сложность;
 - логарифмическая сложность.
3. Затраты аппаратных ресурсов, в основном – памяти:
 - требующие дополнительных затрат памяти;
 - почти не требующие дополнительных затрат памяти.
4. Число проходов по массиву:
 - однопроходные;
 - многопроходные.
5. Устойчивость:
 - меняющие местами одинаковые элементы;
 - не меняющие местами одинаковые элементы.

Сортировка вставками (вариант 1)

Порядок выполнения шагов алгоритма:

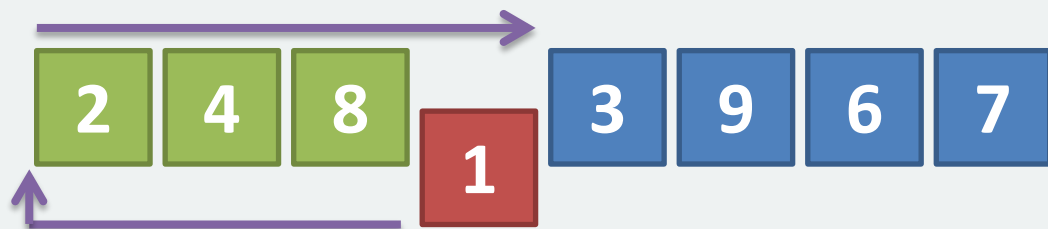
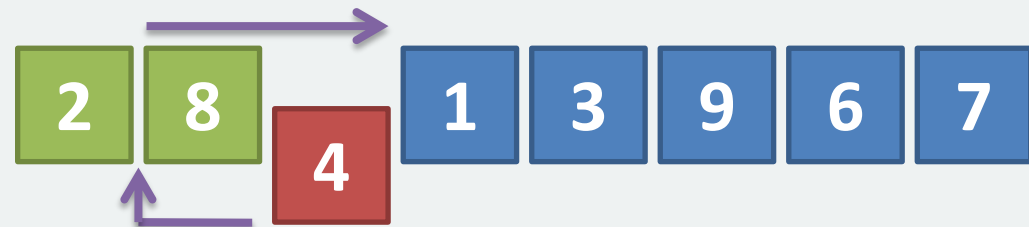
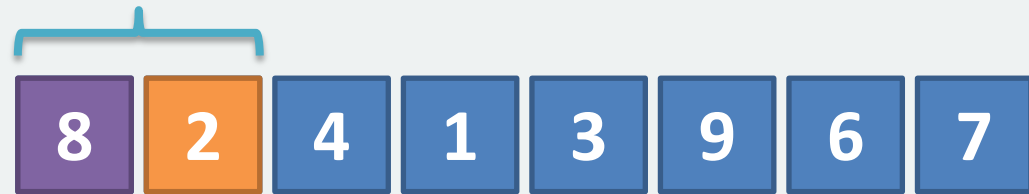
1. задать N равным размеру массива;
2. найти среди N элементов минимальный;
3. вставить его на первое из N место, остальные элементы сдвинуть вправо;
4. уменьшить N на единицу, сдвинув левую границу вправо
5. повторять пункты 2 - 4 до тех пор, пока $N > 0$



Сортировка вставками (вариант 2)

Порядок выполнения шагов алгоритма:

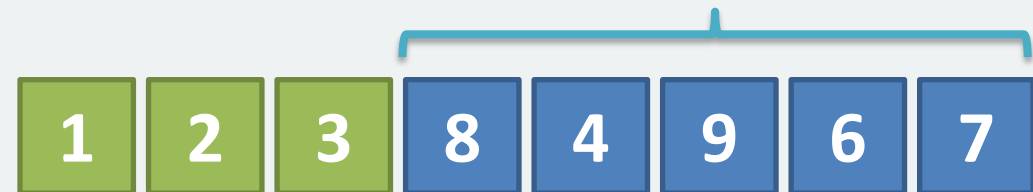
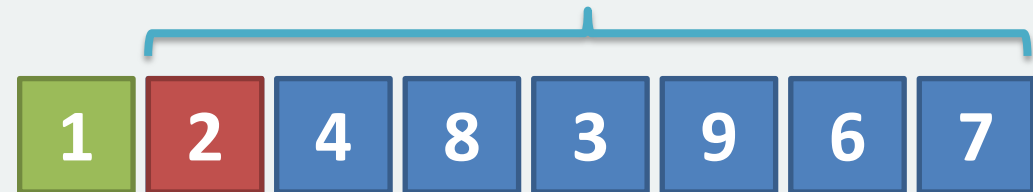
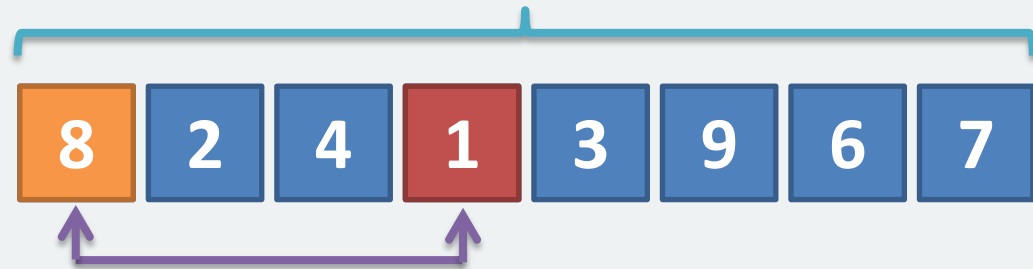
1. выбираем начальный диапазон N в два элемента;
2. сравниваем и сортируем первые два элемента (весь выбранный диапазон);
3. расширяем диапазон N на 1 ячейку вправо;
4. вставляем новый элемент на его место по порядку в отсортированном массиве, смещаем остальные вправо;
5. повторять пункты 3, 4 до тех пор, пока диапазон не равен длине массива.



Сортировка выбором: модификация сортировки вставками

Порядок выполнения шагов алгоритма:

1. задаём начальный диапазон N равным длине массива;
2. среди N элементов ищем минимальный;
3. меняем его местами с первым элементом из диапазона N ;
4. сужаем диапазон N на 1 ячейку вправо;
5. повторяем пункты 2 - 4 до тех пор, пока $N > 0$.



Пузырьковая сортировка

Шаги одного прохода сортировки обменом:



1. начать со второго элемента массива,
 $j = 1$;

2. сравнить j -тый элемент с
предыдущим;

3. если текущий элемент меньше
предыдущего, обменять их местами;

4. перейти к следующему элементу,
увеличив j на единицу;

5. повторять пункты 2 – 4, пока j не
равно размеру массива;

6. повторить пункты 1-5 число раз,
равное размеру массива.

```
for (i = 0; i < mas.length; ++i) {
```

```
    for (j = 1; j < mas.length; ++j) {  
        if (a[j] < a[j - 1])  
            swap(a[j], a[j - 1]);  
    }
```

```
}
```

Модификация пузырьковой сортировки: шейкерная сортировка

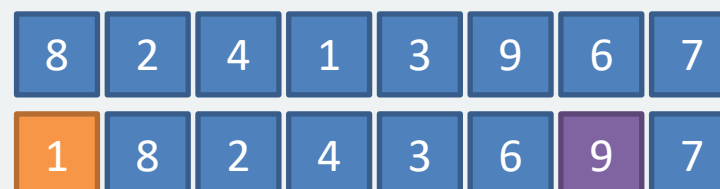
Пузырьковая слева направо

(результат работы первого прохода)



Пузырьковая слева направо

(результат работы первого прохода)



Шейкерная (направо-налево)

(результат работы первого прохода)



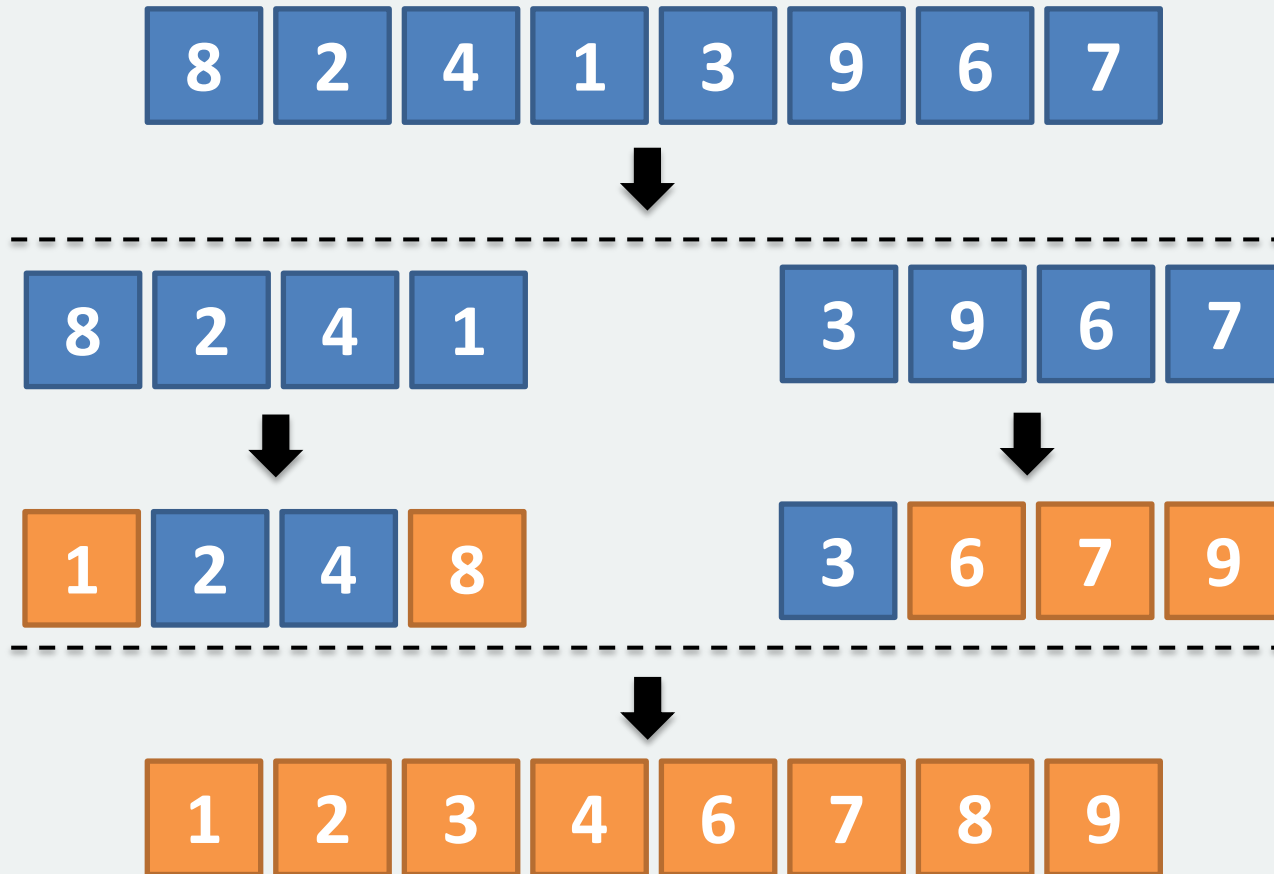
Сортировка слиянием: «разделяй и властвуй»



Порядок выполнения шагов:

1. массив делится на равные (по возможности) части;
2. каждая из минимальных подгрупп сортируется;
3. отсортированные подгруппы сливаются и снова сортируются.

Сортировка слиянием



Быстрая сортировка (Хоара)



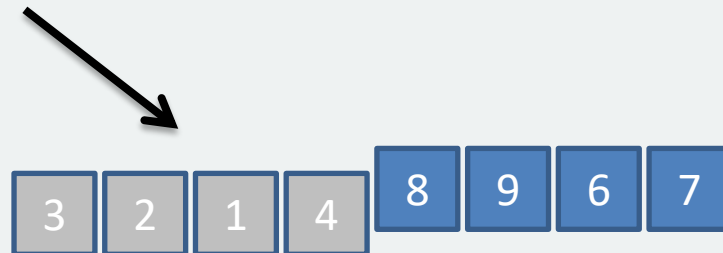
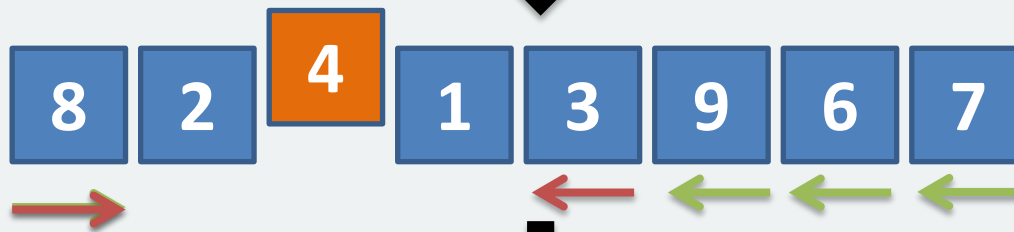
Порядок выполнения шагов:

1. находится некоторый опорный элемент, медиана, массив относительно него делится на две части;
2. все элементы перегруппировываются по отношению к опорному: слева от опорного оказываются элементы меньше его, справа - больше;
3. над каждой из частей массива производится операции 1-3.

Быстрая сортировка (Хоара)



Чарльз Энтони
Ричард Хоар



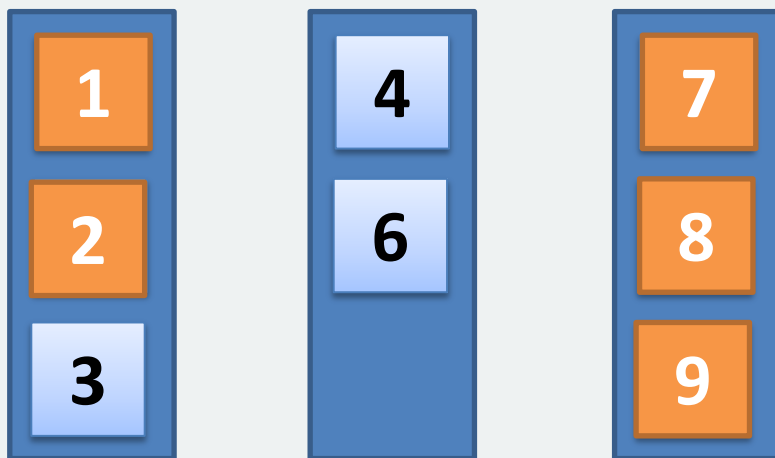
Блочная сортировка



Порядок выполнения шагов:

1. для массива определяются несколько интервалов значений;
2. данные из массива распределяются в дополнительную память в соответствии со своими значениями;
3. данные в дополнительной памяти сортируются;
4. заполняется исходный массив.

Блочная сортировка



Выделим 3 блока с интервалами:

- 1-3
- 4-6
- 7-9

Заполним блоки значениями в соответствии с интервалами

Отсортируем содержимое блоков

Запишем результат в исходный массив

Обработка линейных данных: сортировка подсчётом

Алгоритм эффективен в том случае, когда нужно



отсортировать большое число значений из маленького диапазона.

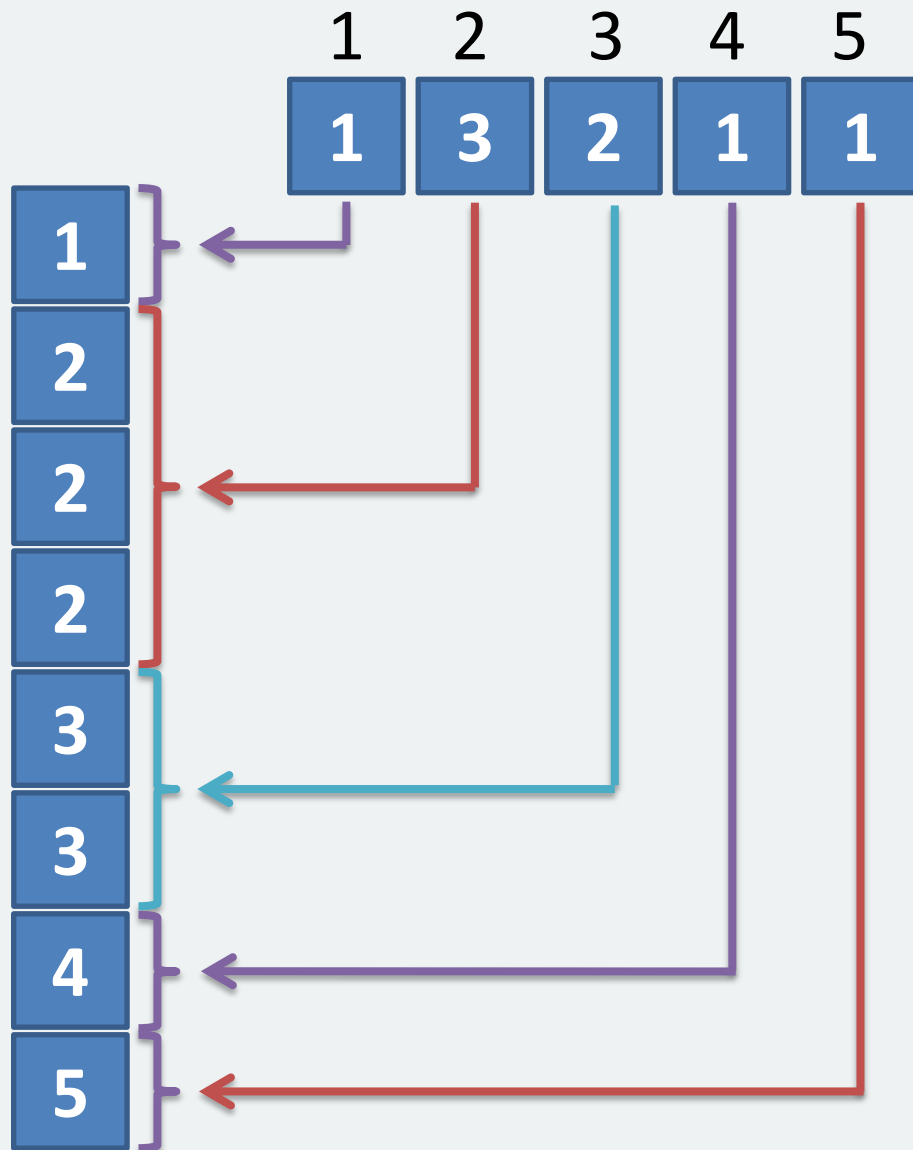
Порядок выполнения шагов:

1. создать дополнительный массив по размеру диапазона, обнулить все элементы;
2. пробежаться по основному массиву, инкрементируя соответствующие значения из дополнительного массива;
3. заполнить исходный массив в соответствии с данными дополнительного массива.

Сортировка подсчётом: сортировка массива

3	2	3	2	2	1	4	5
0	0	0	0	0			

0	0	1	0	0
0	1	1	0	0
0	1	2	0	0
0	2	2	0	0
0	3	2	0	0
1	3	2	0	0
1	3	2	1	0
1	3	2	1	1



Алгоритмы на строках: поиск

Алгоритмы поиска

```
graph TD; A[Алгоритмы поиска] --> B[Поиск подстроки в строке]; A --> C[Поиск среди строк]; B --> D[Точность поиска в строках]; C --> D; D --> E[Поиск точного соответствия]; D --> F[Поиск неточного соответствия];
```

Поиск подстроки в строке

Поиск среди строк

Точность поиска в строках

Поиск точного соответствия

Поиск неточного соответствия

Алгоритм КМП: префикс и суффикс

Подстрока для поиска:

a b c a b d

$\pi = () () () () () ()$

Префикс длины 1 : «a»

Длина префикса должна быть меньше длины подстроки.

Префикс длины 2 : «a», «b»

Суффикс длины 1 : «d»

Длина суффикса должна быть меньше длины подстроки.

Суффикс длины 2 : «b», «d»

Алгоритм КМП: формирование массива « π » (1)

Подстрока для поиска:



$$\pi = (0) () () () () ()$$

Важно!

1. Алгоритм рассматривает только префиксы и суффиксы **одинаковой** длины!
2. Алгоритм рассматривает только ситуации, когда префикс **равен** суффиксу (лексикографически)!

Правило:

$\pi[i]$ равен максимальной длине совпадающих префикса и суффикса в подстроке длины i

Алгоритм КМП: формирование массива « π » (2)

Подстрока для поиска:



префикс длины 1

суффикс длины 1

$$\pi = (0)(0)(\quad)(\quad)(\quad)(\quad)$$

Алгоритм КМП: формирование массива «л» (3)

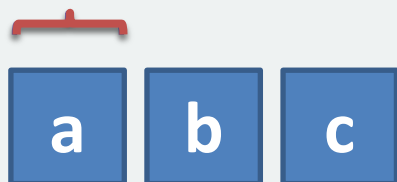
Подстрока для поиска:



$$\pi = (0)(0)(0)(\quad)(\quad)(\quad)$$

Варианты префиксов и суффиксов:

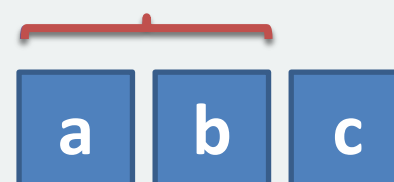
Префиксы



Суффиксы



Префиксы



Суффиксы



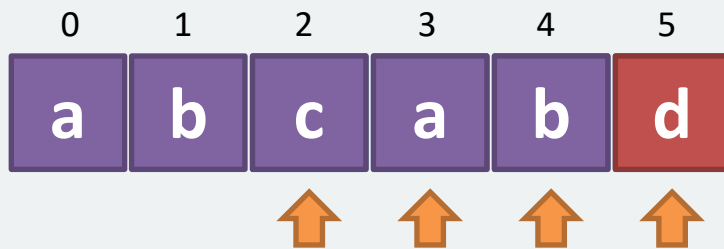
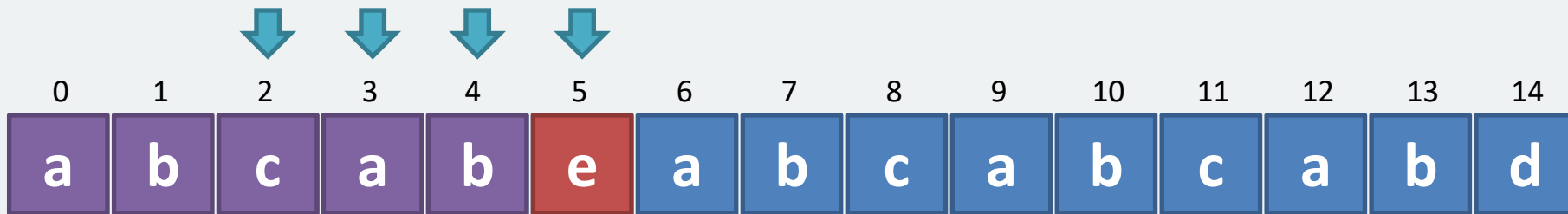
Алгоритм КМП: формирование массива « π » (6)

Подстрока для поиска:

a b c a b d

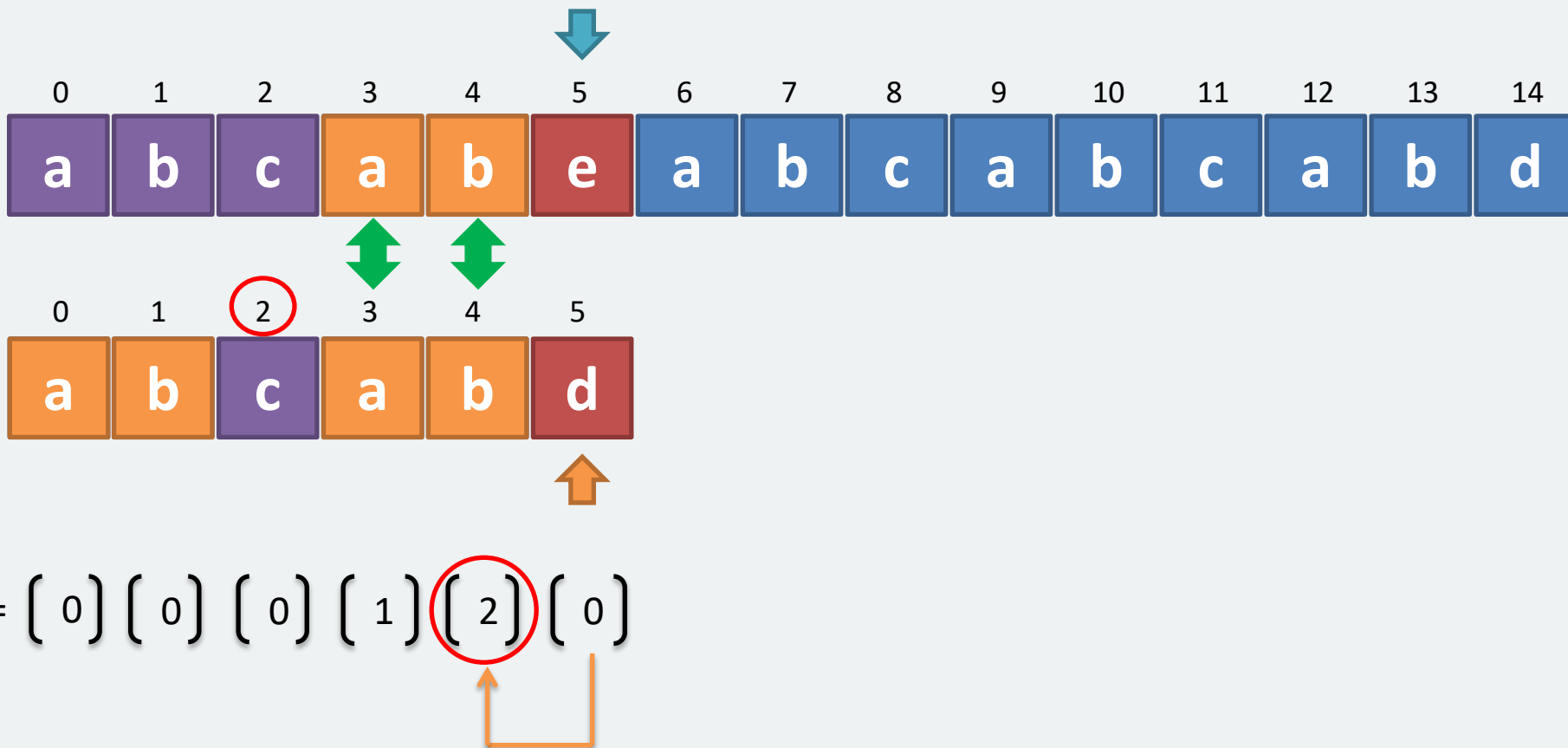
$$\pi = (0)(0)(0)(1)(2)(0)$$

Алгоритм КМП: поиск подстроки в строке (3)

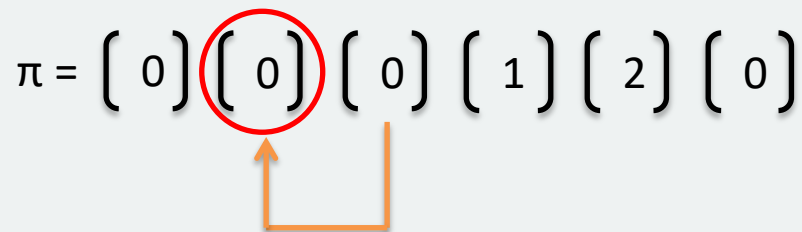
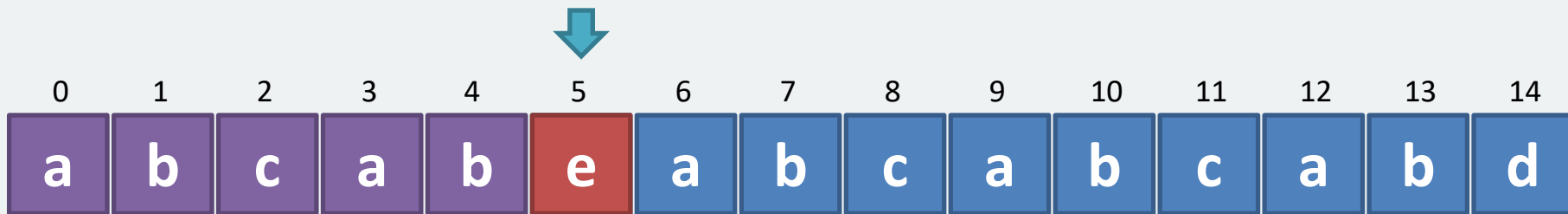


$$\pi = \left(\begin{array}{c} 0 \\ \end{array} \right) \left(\begin{array}{c} 0 \\ \end{array} \right) \left(\begin{array}{c} 0 \\ \end{array} \right) \left(\begin{array}{c} 1 \\ \end{array} \right) \left(\begin{array}{c} 2 \\ \end{array} \right) \left(\begin{array}{c} 0 \\ \end{array} \right)$$

Алгоритм КМП: поиск подстроки в строке (4)



Алгоритм КМП: поиск подстроки в строке (5)



Поиск среди набора строк: префиксный список (бор)

просп центральный 201
просп центральный 301
просп центральный 302
ул юности 6
ул юности 7
ул юности 8
ул юности 9
ул юности 71
ул юности 72

