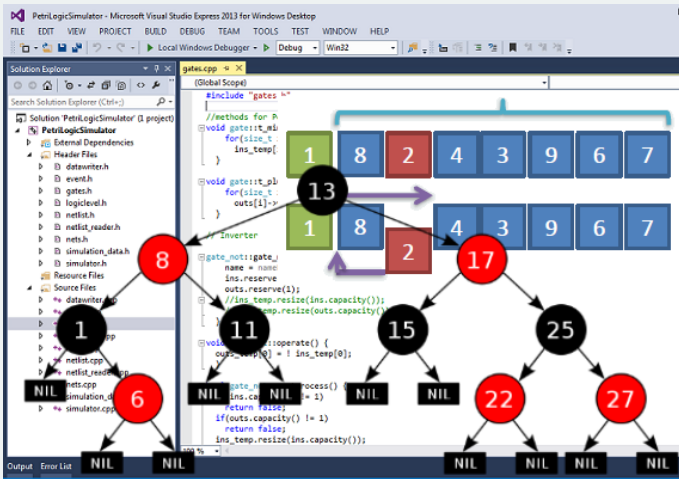




Теория алгоритмов

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, остальные смещаем вправо;
5. повторять пункты 3, 4 до тех пор, пока диапазон N не равен длине массива.

Сортировка выбором: модификация сортировки вставками

8 2 4 1 3 9 6 7

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 $N > 0$.

Сортировка выбором vs. Сортировка вставками

Сортировка вставками

Очень большое число перемещений элементов массива, медленно работает на массивах

Высокая алгоритмическая сложность, составляет порядка N^2

Алгоритм эффективен при работе со списками

Сортировка выбором

Перемещение элементов массива сведено к минимуму, вместо перемещения введён обмен элементов

Высокая алгоритмическая сложность, составляет порядка N^2

Алгоритм эффективен при работе со списками и массивами

Сортировки обменом: основная идея

8 2 4 1 3 9 6 7

Основная идея сортировок обменом – отказ от перемещения большого числа элементов массива на далёкие расстояния.

Шаги одного прохода сортировки обменом:

1. начать со второго элемента массива, $j = 1$;
2. сравнить текущий j -тый элемент с предыдущим;
3. если текущий элемент меньше предыдущего, поменять их местами;
4. перейти к следующему элементу, увеличив j на единицу;
5. повторять пункты 2 – 4, пока j не равно размеру массива;

Пузырьковая сортировка

Шаги одного прохода сортировки обменом:



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

2. сравнить текущий j -тый элемент с
предыдущим;

3. если текущий элемент меньше
предыдущего, поменять 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]);  
    }
```

```
}
```

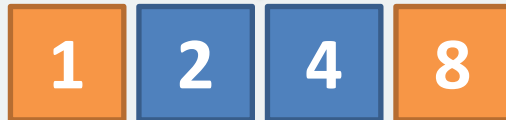
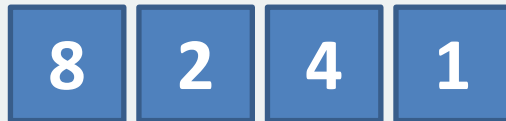

Сортировка слиянием: «разделяй и властвуй»



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: «разделяй и властвуй»



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.

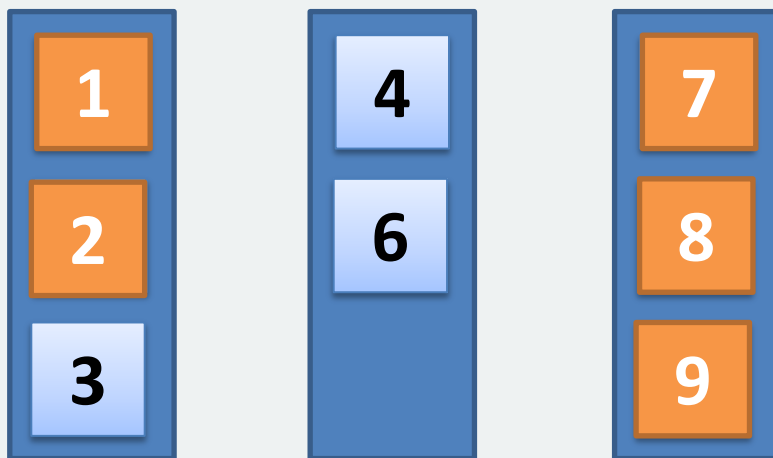
Блочная сортировка



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:

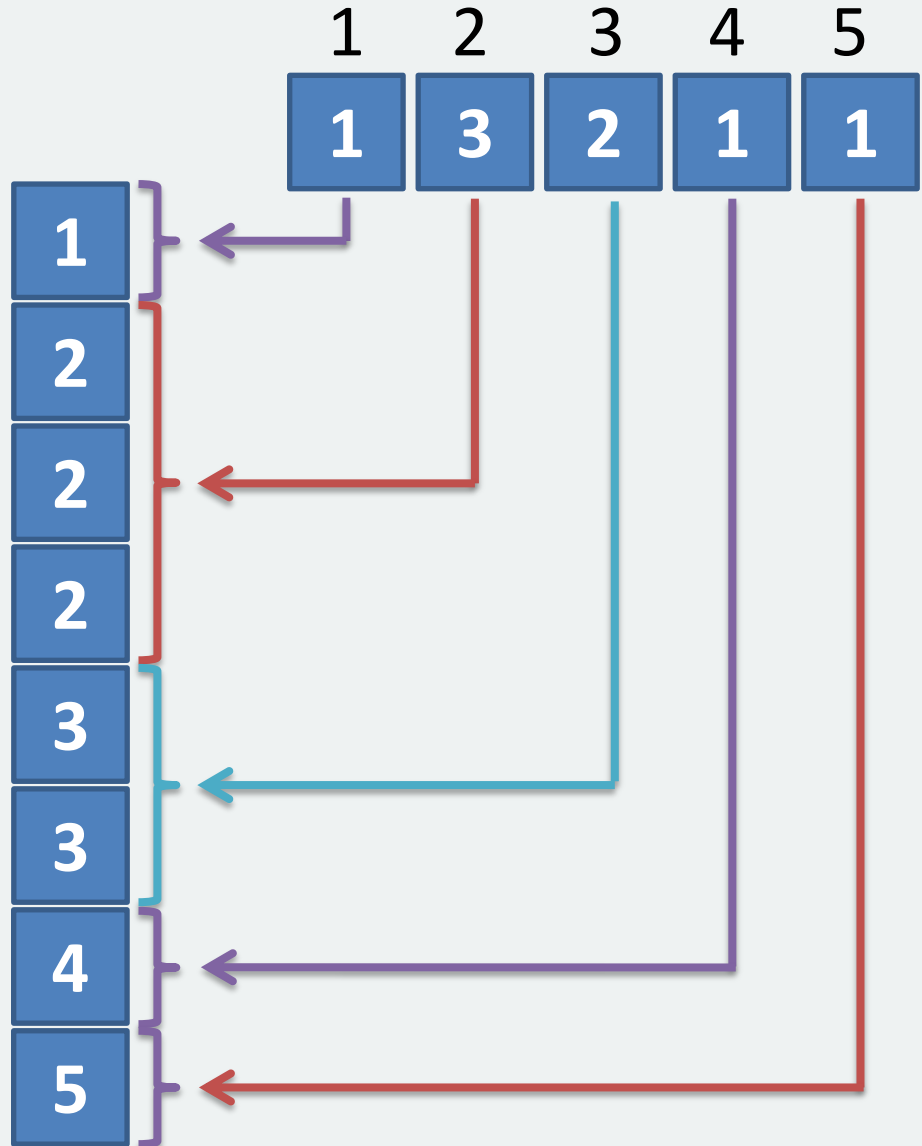
1. создать дополнительный массив по размеру диапазона, обнулить все элементы;
2. пробежаться по основному массиву, инкрементируя соответствующие значения из дополнительного массива;
3. заполнить исходный массив в соответствии с данными дополнительного массива.



Сортировка подсчётом: сортировка массива

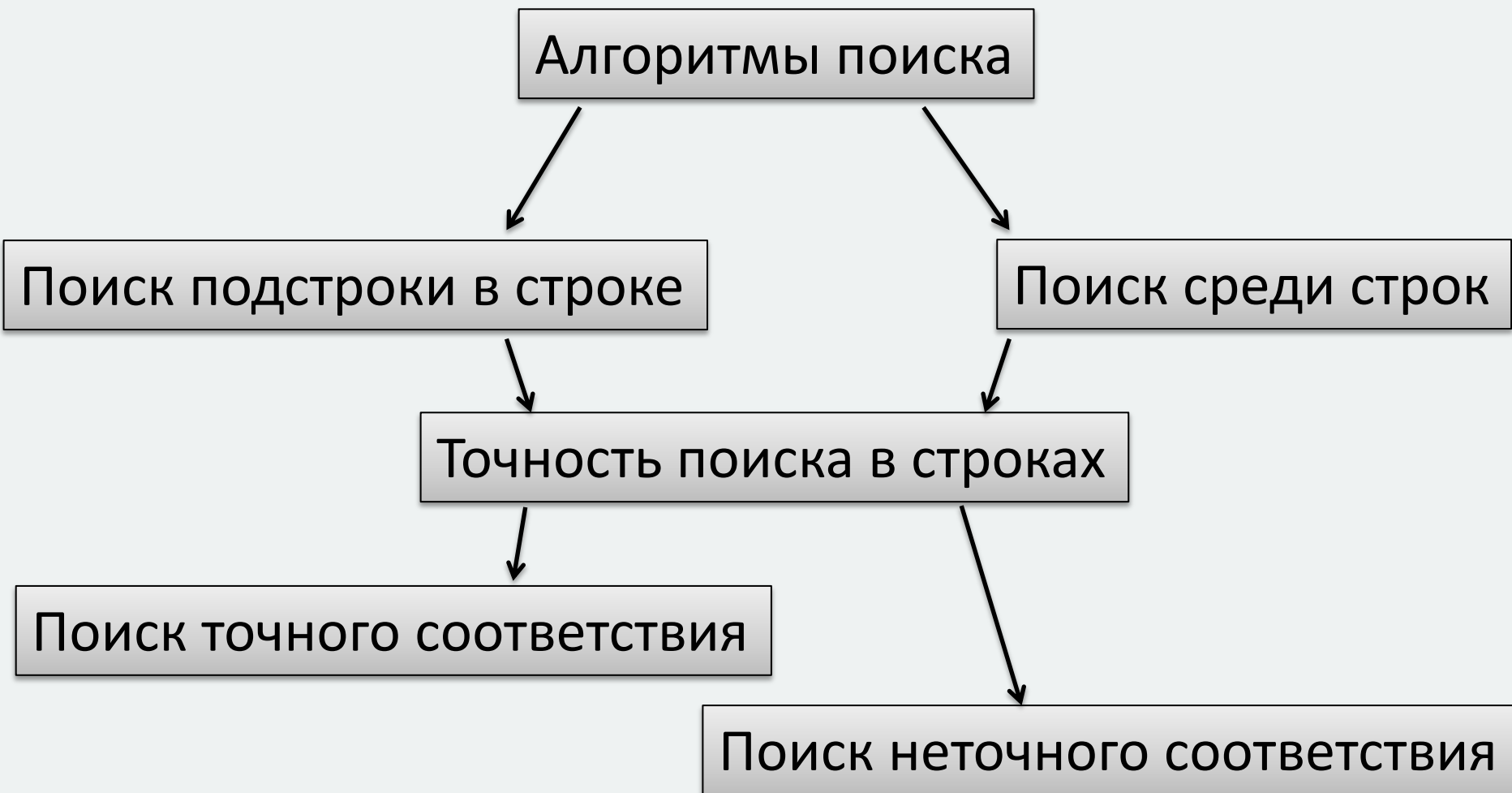
3	2	3	2	2	1	4	5
0	0	0	0	0			

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



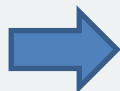


Алгоритмы на строках: поиск



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



На хранение этих 7-ми
строк тратится всего
39 байт
вместо 107 для строк

