Одномерные массивы: задачи сортировок элементов массива

АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ

Цели и задачи лекции

Цель лекции – изучить понятия и классификацию алгоритмов сортировок массивов, реализацию алгоритмов простых сортировок и научиться решать задачи на сортировку одномерных массивов с помощью алгоритмов простых сортировок в языке C++.

Определения

Сортировка – это упорядочивание набора однотипных данных по возрастанию или убыванию.

Сортировка применяется для облегчения поиска элементов в упорядоченном множестве. Задача сортировки одна из фундаментных в программировании.

В общем случае сортировку следует понимать как процесс перегруппировки заданного множества объектов в определенном порядке.

Оценка алгоритмов сортировки

Существует множество различных алгоритмов сортировки. Все они имеют свои положительные и отрицательные стороны. Перечислим общие критерии оценки алгоритмов сортировки:

- **Скорость работы алгоритма сортировки.** Она непосредственно связана с количеством сравнений и количеством обменов, происходящих во время сортировки, причем обмены занимают больше времени.

Оценка алгоритмов сортировки

Сравнение происходит тогда, когда один элемент массива сравнивается с другим; обмен происходит тогда, когда два элемента меняются местами. Время работы одних алгоритмов сортировки растет экспоненциально, а время работы других логарифмически зависит от количества элементов.

- **Время работы в лучшем и худшем случаях.** Оно имеет значение при анализе выполнения алгоритма, если одна из краевых ситуаций будет встречаться довольно часто.

Оценка алгоритмов сортировки

Алгоритм сортировки зачастую имеет хорошее среднее время выполнения, но в худшем случае он работает очень медленно.

- **Поведение алгоритма сортировки.** Поведение алгоритма сортировки называется естественным, если время сортировки минимально для уже упорядоченного списка элементов, увеличивается по мере возрастания степени неупорядоченности списка и максимально, когда элементы списка расположены в обратном порядке.

Оценка алгоритмов сортировки

Различные сортировки массивов отличаются по быстродействию. Существуют простые методы сортировок, которые требуют порядка n*n сравнений, где n – количество элементов массива и быстрые сортировки, которые требуют порядка n*ln(n) сравнений. Простые методы удобны для объяснения принципов сортировок, т.к. имеют простые и короткие алгоритмы. Усложненные методы требуют меньшего числа операций, но сами операции более сложные, поэтому для небольших массивов простые методы более эффективны.

Оценка алгоритмов сортировки

Простые методы сортировки можно разделить на три основные категории:

- сортировка методом "пузырька" (простого обмена);
- сортировка методом простого выбора (простой перебор);
- сортировка методом простого включения (сдвиг-вставка, вставками, вставка и сдвиг).

Оценка алгоритмов сортировки

Простые методы сортировки можно разделить на три основные категории:

- сортировка методом "пузырька" (простого обмена);
- сортировка методом простого выбора (простой перебор);
- сортировка методом простого включения (сдвиг-вставка, вставками, вставка и сдвиг).

Сортировка методом "пузырька" (простого обмена)

Самый известный алгоритм – пузырьковая сортировка (bubble sort, сортировка методом пузырька или просто сортировка пузырьком). Его популярность объясняется интересным названием и простотой самого алгоритма.

Алгоритм попарного сравнения элементов массива в литературе часто называют "методом пузырька", проводя аналогию с пузырьком, поднимающимся со дна бокала с газированной водой. По мере всплывания пузырек сталкивается с другими пузырьками и, сливаясь с ними, увеличивается в

Сортировка методом "пузырька" (простого обмена)

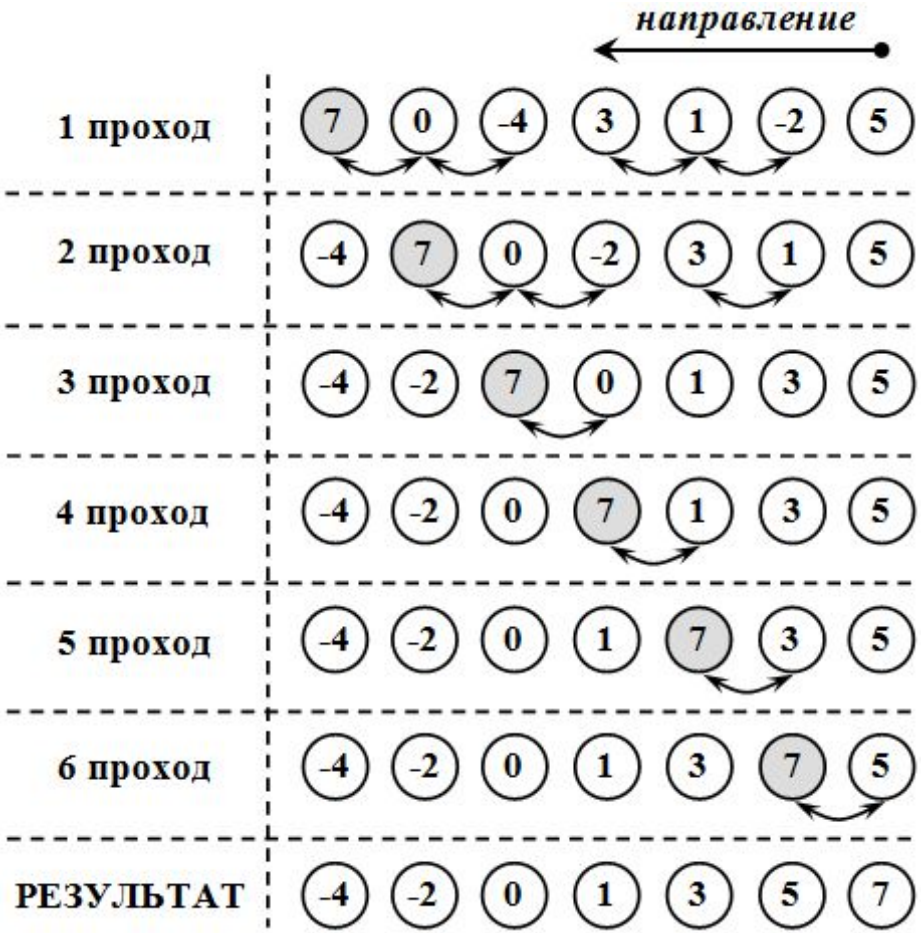
Чтобы аналогия стала очевидной, нужно считать, что элементы массива расположены вертикально друг над другом, и их нужно так упорядочить, чтобы они увеличивались сверху вниз.

Алгоритм состоит в повторяющихся проходах по сортируемому массиву. За каждый проход элементы последовательно сравниваются попарно и, если порядок в паре неверный, выполняется обмен элементов.

Сортировка методом "пузырька" (простого обмена)

Проходы по массиву повторяются до тех пор, пока на очередном проходе не окажется, что обмены больше не нужны, что означает – массив отсортирован. При проходе алгоритма элемент, стоящий не на своём месте, "всплывает" до нужной позиции.

Демонстрация сортировки по неубыванию методом "пузырька"



Сортировка методом "пузырька" (простого обмена)

```
//Описание функции сортировки методом "пузырька"
void BubbleSort (int k,int x[max]) {
  int i,j,buf;
  for (i=k-1;i>0;i--)
    for (j=0;j<i;j++)
    if (x[j]>x[j+1]) {
      buf=x[j];
```

Сортировка методом "пузырька" (простого обмена)

```
x[j]=x[j+1];
     x[j+1]=buf;
    }
}
```

Сортировка методом "пузырька" (простого обмена)

В пузырьковой сортировке количество сравнений всегда одно и то же, поскольку два цикла for повторяются указанное количество раз независимо от того, был список изначально упорядочен или нет. Это значит, что алгоритм пузырьковой сортировки всегда выполняет $\frac{(n^2-n)}{2}$ сравнений, где n – количество сортируемых элементов. Данная формула выведена на том основании, что внешний цикл выполняется n-1 раз, а внутренний выполняется в среднем n/2 раз.

Сортировка методом "пузырька" (простого обмена)

Пузырьковая сортировка имеет такую особенность: неупорядоченные элементы на "большом" конце массива занимают правильные положения за один проход, но неупорядоченные элементы в начале массива поднимаются на свои места очень медленно. Поэтому, вместо того чтобы постоянно просматривать массив в одном направлении, в последовательных проходах можно чередовать направления. Таким образом, элементы, сильно удаленные от своих положений, быстро станут на свои места.

Сортировка методом "пузырька" (простого обмена)

Данная версия пузырьковой сортировки носит название **шейкер-сортировки** (shaker sort сортировка перемешиванием, сортировка взбалтыванием, сортировка встряхиванием), поскольку действия, производимые ею с массивом, напоминают взбалтывание или встряхивание. Ниже показана реализация шейкер-сортировки.

Сортировка методом "пузырька" (простого обмена)

```
//Описание функции шейкер-сортировки
void Shaker(int k,int x[max]){
  int i,t;
  bool exchange;
  do {
    exchange = false;
    for(i=k-1; i > 0; --i) {
      if(x[i-1] > x[i]) {
```

Сортировка методом "пузырька" (простого обмена)

```
    t = x[i-1];
    x[i-1] = x[i];
    x[i] = t;
    exchange = true;
  }
}
for(i=1; i < k; ++i) {
  if(x[i-1] > x[i]) {
```

Сортировка методом "пузырька" (простого обмена)

```
        t = x[i-1];
        x[i-1] = x[i];
        x[i] = t;
        exchange = true;
      }
    }
  } while(exchange);
    //сортировать до тех пор, пока не будет обменов
}
```

Сортировка методом "пузырька" (простого обмена)

В пузырьковой сортировке количество сравнений всегда одно и то же, поскольку два цикла for повторяются указанное количество раз независимо от того, был список изначально упорядочен или нет. Это значит, что алгоритм пузырьковой сортировки всегда выполняет $\frac{(n^2-n)}{2}$ сравнений, где n – количество сортируемых элементов. Данная формула выведена на том основании, что внешний цикл выполняется n-1 раз, а внутренний выполняется в среднем n/2 раз.

Сортировка методом простого выбора (простой перебор)

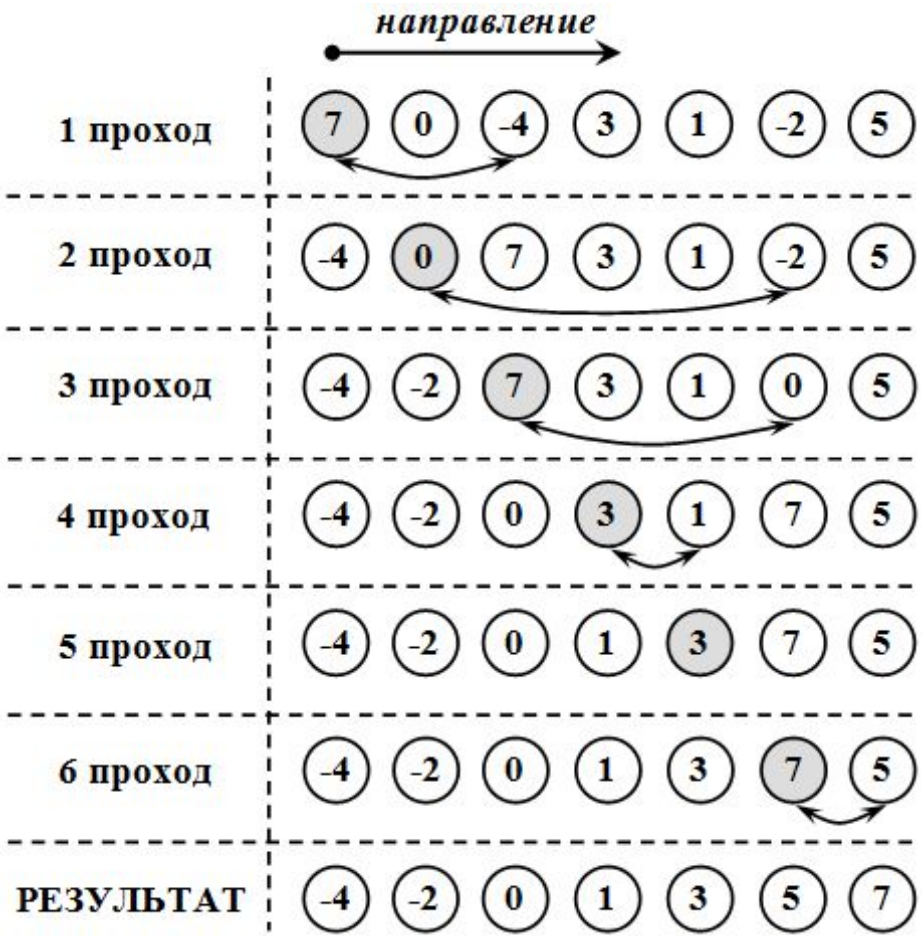
Это наиболее естественный алгоритм упорядочивания. При данной сортировке из массива выбирается элемент с наименьшим значением и обменивается с первым элементом. Затем из оставшихся n - 1 элементов снова выбирается элемент с наименьшим ключом и обменивается со вторым элементом, и т.д.

Сортировка методом простого выбора (простой перебор)

Шаги алгоритма:

- находим минимальное значение в текущей части массива;
- производим обмен этого значения со значением на первой неотсортированной позиции;
- далее сортируем хвост массива, исключив из рассмотрения уже отсортированные элементы.

Демонстрация сортировки по неубыванию методом простого выбора



Сортировка методом простого выбора (простой перебор)

```
//Описание функции сортировки методом простого выбора
void SelectionSort (int k,int x[max]) {
  int i,j,min,temp;
  for (i=0;i<k-1;i++) {
  //устанавливаем начальное значение минимального индекса
    min=i;
```

Сортировка методом простого выбора (простой перебор)

```
//находим минимальный индекс элемента
   for (j=i+1;j<k;j++){
     if (x[j]<x[min])
       min=j;
     //меняем значения местами      }
     temp=x[i];
     x[i]=x[min];
     x[min]=temp;
  }
}
```

Сортировка методом простого выбора (простой перебор)

В пузырьковой сортировке количество сравнений всегда одно и то же, поскольку два цикла for повторяются указанное количество раз независимо от того, был список изначально упорядочен или нет. Это значит, что алгоритм пузырьковой сортировки всегда выполняет $\frac{(n^2-n)}{2}$ сравнений, где n – количество сортируемых элементов. Данная формула выведена на том основании, что внешний цикл выполняется n-1 раз, а внутренний выполняется в среднем n/2 раз

Сортировка методом простого включения (сдвиг-вставка, вставками, вставка и сдвиг)

Хотя этот метод сортировки намного менее эффективен, чем сложные алгоритмы (такие как быстрая сортировка), у него есть ряд преимуществ:

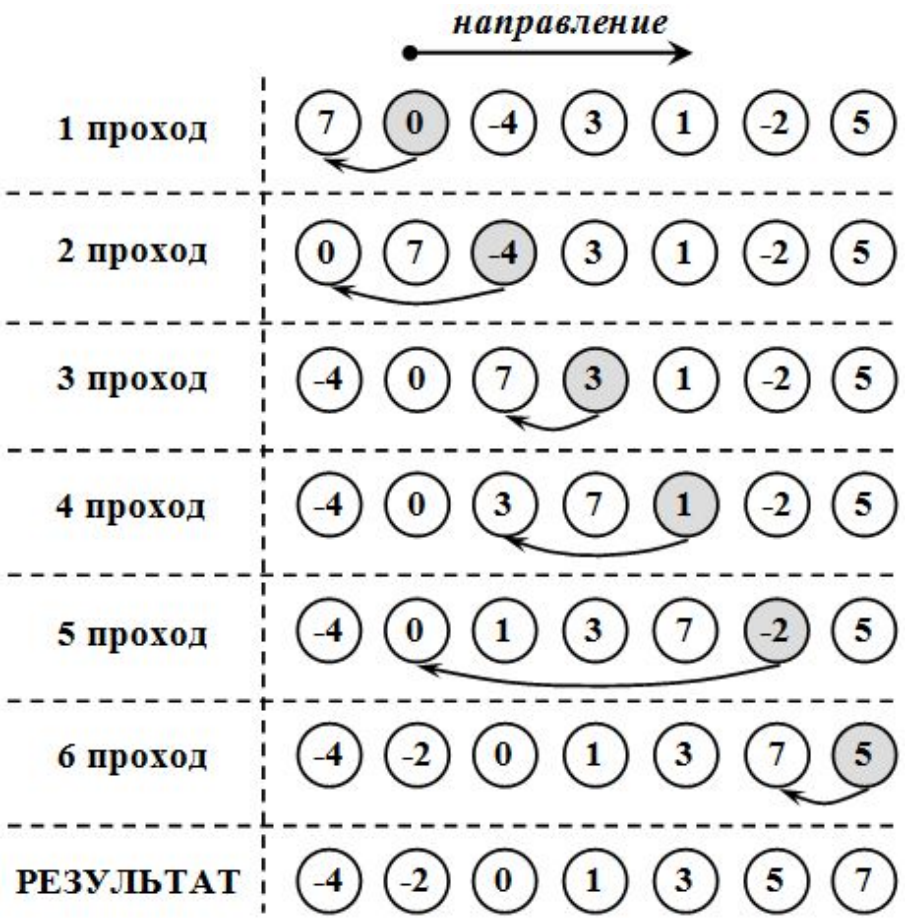
- прост в реализации;
- эффективен на небольших наборах данных, на наборах данных до десятков элементов может оказаться лучшим;
- эффективен на наборах данных, которые уже частично отсортированы;
- это устойчивый алгоритм сортировки (не меняет порядок элементов, которые уже отсортированы);

Сортировка методом простого включения (сдвиг-вставка, вставками, вставка и сдвиг)

- может сортировать массив по мере его получения;
- не требует временной памяти, даже под стек.

На каждом шаге алгоритма выбираем один из элементов входных данных и вставляем его на нужную позицию в уже отсортированной последовательности до тех пор, пока набор входных данных не будет исчерпан. Метод выбора очередного элемента из исходного массива произволен; может использоваться практически любой алгоритм выбора.

Демонстрация сортировки по неубыванию методом простого включения



Сортировка методом простого включения (сдвиг-вставка, вставками, вставка и сдвиг)

```
//Описание функции сортировки методом простого включения
void InsertSort (int k,int x[max]) {
  int i,j, temp;
  for (i=0;i<k;i++) {
  //цикл проходов, i - номер прохода
    temp=x[i];
```

Сортировка методом простого включения (сдвиг-вставка, вставками, вставка и сдвиг)

```
//поиск места элемента
    for (j=i-1; j>=0 && x[j]>temp; j--)
      x[j+1]=x[j];//сдвигаем элемент вправо, пока не дошли
      //место найдено, вставить элемент
      x[j+1]=temp;
  }
}
```

Сортировка методом простого выбора (простой перебор)

В пузырьковой сортировке количество сравнений всегда одно и то же, поскольку два цикла for повторяются указанное количество раз независимо от того, был список изначально упорядочен или нет. Это значит, что алгоритм пузырьковой сортировки всегда выполняет $\frac{(n^2-n)}{2}$ сравнений, где n – количество сортируемых элементов. Данная формула выведена на том основании, что внешний цикл выполняется n-1 раз, а внутренний выполняется в среднем n/2 раз.

Спасибо за внимание