ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ВЫСШАЯ ШКОЛА ЭКОНОМИКИ

Сортировки

Задача сортировки

Пусть есть последовательность $a_0, a_1 \ldots a_n$ и функция сравнения, которая на любых двух элементах последовательности принимает одно из трех значений: меньше, больше или равно.

Задача сортировки состоит в перестановке членов последовательности таким образом, чтобы выполнялось условие:

$$a_i <= a_{i+1}, \qquad \text{для всех } i \text{ от } 0 \text{ до } n.$$

Пожалуй, никакая другая проблема не породила такого количества разнообразнейших решений, как задача сортировки. Существует ли некий "универсальный", наилучший алгоритм?

Имея приблизительные характеристики входных данных, можно подобрать метод, работающий оптимальным образом.

Параметры алгоритмов сортировки

1) Время сортировки - основной параметр, характеризующий быстродействие алгоритма.

2) Память - ряд алгоритмов требует выделения дополнительной памяти под временное хранение данных. При оценке используемой памяти не будет учитываться место, которое занимает исходный массив и не зависящие от входной последовательности затраты, например, на хранение кода программы.

3) Устойчивость - устойчивая сортировка не меняет взаимного расположения равных элементов. Такое свойство может быть очень полезным, если они состоят из нескольких полей, а сортировка происходит по одному из них.

Сфера применения

1) внутренние сортировки работают с данными в оперативной памяти с произвольным доступом;

2) внешние сортировки упорядочивают информацию, расположенную на внешних носителях. Это приводит к специальным методам сортировки, обычно использующим дополнительное дисковое пространство, и накладывает некоторые дополнительные ограничения на алгоритм:

- доступ к носителю осуществляется последовательным образом: в каждый момент времени можно считать или записать только элемент, следующий за текущим;
- объем данных не позволяет им разместиться в ОЗУ. Кроме того, доступ к данным на носителе производится намного медленнее, чем операции с оперативной памятью.

Стратегии сортировки

1) **Стратегия выборки**. Из входного множества выбирается следующий по критерию упорядоченности элемент и включается в выходное множество на место, следующее по номеру.

2) **Стратегия включения**. Из входного множества выбирается следующий по номеру элемент и включается в выходное множество на то место, которое он должен занимать в соответствии с критерием упорядоченности.

3) **Стратегия распределения.** Входное множество разбивается на ряд подмножеств (возможно, меньшего объема) и сортировка ведется внутри каждого такого подмножества.

4) **Стратегия слияния.** Выходное множество получается путем слияния маленьких упорядоченных подмножеств.

Классификация методов сортировки

Сортировка (Sorting) - процесс перегруппировки *однотипных* элементов данных в некотором определенном порядке.

Алгоритмы сортировки можно разделить на

- Итерационные и рекурсивные
- По месту и с использованием дополнительной памяти
- По асимптотической оценке сложности O(n)
- По требованиям к входным данным и т.д.

Важно выбрать тот алгоритм, который лучше всего подходит для решения конкретной задачи.

Все рассматриваемые методы предназначены для сортировки по возрастанию линейных массивов с нижней границей индекса равной 1.

Сортировка – от чего зависит скорость

У каждого алгоритма сортировки есть свои преимущества и недостатки.
Производительность различных алгоритмов зависит от

- типа данных,
- начального расположения данных (массив «почти сортирован», сортирован в обратном порядке, данные случайно расположены)
- размера массива
- значений элементов массива (массив заполнен значениями из небольшого / большого диапазона)и т.д.

Основные идеи итерационных алгоритмов сортировки

1. ***Обмен***. Если два элемента расположены не в порядке возрастания, они меняются местами
2. ***Сортировка посредством выбора***. Выбираем наименьший ставим на первое место. Среди оставшихся снова выбираем наименьший и ставим на второе место. И.т.д.
3. ***Вставка***. Элементы просматриваются по одному и каждый новый элемент вставляется в подходящее место среди ранее упорядоченных
4. ***Сортировка путем подсчета***. Для каждого элемента считаем количество меньших элементов – это количество и будет его позицией в отсортированном массиве.

Договоримся

Алгоритмы задаются

- Блок-схемой
- на псевдокоде,
- на языке программирования

В алгоритмах на псевдокоде отступ от левого поля указывает на уровень вложенности для составных операторов (условных и циклов), специальные команды для обозначения начала и конца блока не используются.

В алгоритмах на псевдокоде тело цикла

```
repeat … until <логическое выражение>
```

выполняется, когда `<логическое выражение>` ложно

В алгоритмах на псевдокоде индексы массивов изменяются от 1 до N. Учитывайте при написании кода на языке программирования