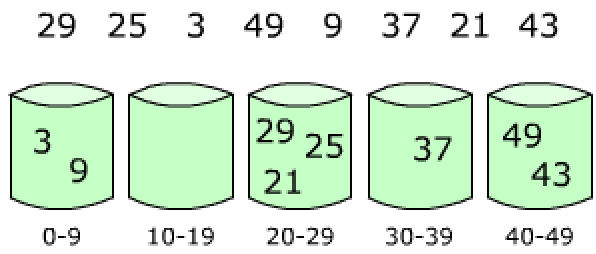
ВикипедиЯ

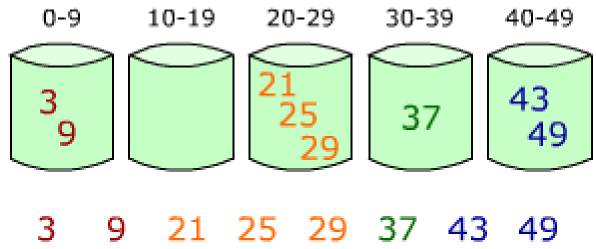
Блочная сортировка

Материал из Википедии — свободной энциклопедии

Блочная сортировка (Карманная сортировка, корзинная сортировка, англ. *Bucket sort*) — алгоритм сортировки, в котором сортируемые элементы распределяются между конечным числом отдельных блоков (карманов, корзин) так, чтобы все элементы в каждом следующем по порядку блоке были всегда больше (или меньше), чем в предыдущем. Каждый блок затем сортируется отдельно, либо рекурсивно тем же методом, либо другим. Затем элементы помещаются обратно в массив. Этот тип сортировки может обладать линейным временем исполнения.



Элементы распределяются по корзинам



Затем элементы в каждой корзине сортируются

Данный алгоритм требует знаний о природе сортируемых данных, выходящих за рамки функций "сравнить" и "поменять местами", достаточных для сортировки слиянием, сортировки пирамидой, быстрой сортировки, сортировки Шелла, сортировки вставкой.

Преимущества: относится к классу быстрых алгоритмов с линейным временем исполнения O(N) (на удачных входных данных).

Недостатки: сильно деградирует при большом количестве мало отличных элементов, или же на неудачной функции получения номера корзины по содержимому элемента. В некоторых таких случаях для строк, возникающих в реализациях основанного на сортировке строк алгоритма сжатия BWT, оказывается, что быстрая сортировка строк в версии Седжвика значительно превосходит блочную сортировку скоростью.

Содержание

Алгоритм

Если входные элементы подчиняются равномерному закону распределения, то математическое ожидание времени работы алгоритма карманной сортировки является линейным. Это возможно благодаря определенным предположениям о входных данных. При карманной сортировке предполагается, что входные данные равномерно распределены на отрезке [0, 1).

Идея алгоритма заключается в том, чтобы разбить отрезок $[0, 1)$ на *n* одинаковых отрезков (карманов), и разделить по этим карманам *n* входных величин. Поскольку входные числа равномерно распределены, предполагается, что в каждый карман попадет небольшое количество чисел. Затем последовательно сортируются числа в карманах. Отсортированный массив получается путём последовательного перечисления элементов каждого кармана.

Псевдокод

```
function bucket-sort(A, n) is
  buckets ← новый массив из n пустых элементов
  for i = 0 to (length(A)-1) do
    вставить A[i] в конец массива buckets[msbits(A[i], k)]
  for i = 0 to n - 1 do
    next-sort(buckets[i])
  return Конкатенация массивов buckets[0], ..., buckets[n-1]
```

На вход функции *bucket-sort* подаются сортируемый массив (список, коллекция и т.п.) *A* и количество блоков - *n*.

Массив *buckets* представляет собой массив массивов (массив списков, массив коллекций и т.п.), подходящих по природе к элементам *A*.

Функция *msbits(x,k)* тесно связана с количеством блоков - *n* (возвращает значение от 0 до n), и, в общем случае, возвращает *k* наиболее значимых битов из *x* (*floor(x/2^(size(x)-k))*). В качестве *msbits(x,k)* могут быть использованы разнообразные функции, подходящие по природе сортируемым данным и позволяющие разбить массив A на n блоков. Например, для символов A-Z это может быть сопоставление номерам букв 0-25, или возврат кода первой буквы (0-255) для ASCII набора символов; для чисел $[0, 1)$ это может быть функция *floor(n*A[i])*, а для произвольного набора чисел в интервале $[a, b)$ - функция *floor(n*(A[i]-a)/(b-a))*.

Функция *next-sort* также реализует алгоритм сортировки для каждого созданного на первом этапе блока. Рекурсивное использование *bucket-sort* в качестве *next-sort* превращает данный алгоритм в поразрядную сортировку. В случае *n = 2* соответствует быстрой сортировке (хотя и с потенциально плохим выбором опорного элемента).

Оценка сложности

Оценим сложность алгоритма блочной сортировки для случая, при котором в качестве алгоритма сортировки блоков (*next-sort* из псевдокода) используется сортировка вставками.

Для оценки сложности алгоритма введём случайную величину n_i, обозначающую количество элементов, которые попадут в карман `B[i]`. Время работы сортировки вставками равно $O\left(n^2\right)$.

Время работы алгоритма карманной сортировки равно

$$T(n) = \Theta(n) + \sum_{i=0}^{n-1} O(n_i^2)$$

Вычислим математическое ожидание обеих частей равенства:

$$M\left(T(n)\right) = M\left(\Theta(n) + \sum_{i=0}^{n-1} O(n_i^2)\right) = \Theta(n) + \sum_{i=0}^{n-1} O\left(M(n_i^2)\right)$$

Найдем величину $M(n_i^2)$.

Введем случайную величину X_{ij}, которая равна 1, если `A[j]` попадает в *i*-й карман, и 0 в противном случае:

$$n_i = \sum_{j=1}^{n} X_{ij}$$

$$M\left(n_i^2\right) = \quad M\left[\left(\sum_{j=1}^n X_{ij}\right)^2\right] = M\left[\sum_{j=1}^n \sum_{k=1}^n X_{ij}X_{ik}\right] = \sum_{j=1}^n M\left[X_{ij}^2\right] + \sum_{1\le j\le n} \ \sum_{1\le k\le n, k\ne j} M\left[X_{ij}X_{ik}\right]$$

$$M\left[X_{ij}^2\right] = 1\cdot\frac{1}{n} + 0\cdot\left(1-\frac{1}{n}\right) = \frac{1}{n}$$

Если $k \neq j$, величины X_{ij} и X_{ik} независимы, поэтому:

$$M\left[X_{ij}X_{ik}\right] = M\left[X_{ij}\right] M\left[X_{ik}\right] = \frac{1}{n^2}$$

Таким образом

$$M\left(n_i^2\right) = \sum_{j=1}^n \frac{1}{n} + \sum_{1\le j\le n} \ \sum_{1\le k\le n, k\ne j} \frac{1}{n^2} = 2 - \frac{1}{n}$$

Итак, ожидаемое время работы алгоритма карманной сортировки равно

$$\Theta(n) + n\cdot O(2-1/n) = \Theta(n)$$

Литература

- *Кормен, Томас Х., Лейзерсон, Чарльз И., Ривест, Рональд Л., Штайн, Клифорд.* **Глава 8. Сортировка за линейное время** // Алгоритмы: построение и анализ, 2-е издание = Introduction to Algorithms second edition. — М.: «Вильямс», 2005. — С. 230 - 234. — ISBN 5-8459-0857-4.

См. также

- Список алгоритмов сортировки

Ссылки

- Визуализатор1 (http://rain.ifmo.ru/cat/view.php/vis/sorts/linear-2005) — Java-аплет.
- Визуализатор2 (http://rain.ifmo.ru/cat/view.php/vis/sorts/linear-2001) — Java-аплет.

Источник — https://ru.wikipedia.org/w/index.php?title=Блочная_сортировка&oldid=90866742

Эта страница в последний раз была отредактирована 11 февраля 2018 в 16:33.

Текст доступен по лицензии Creative Commons Attribution-ShareAlike; в отдельных случаях могут действовать дополнительные условия.
Wikipedia® — зарегистрированный товарный знак некоммерческой организации Wikimedia Foundation, Inc.