# МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

# КИЇВСЬКА ДЕРЖАВНА АКАДЕМІЯ ВОДНОГО ТРАНСПОРТУ

# імені гетьмана Петра-Конашевича Сагайдачного

# 

# КОНТРОЛЬНА РОБОТА

# З ДИСЦИПЛІНИ “ Моделювання інформаційних потоків ”

**студента 5 курсу**

**заочного відділення спеціальності**

**“Менеджмент організацій”**

# К И Ї В − 2 0 0 6

**1. Сортировка вставками**

Пусть надо упорядочить N элементов R1, R2…Rn.

Назовем их записями, а всю совокупность N назовем файлом. Каждая запись Rjимеет свой ключ Kj, который и управляет процессом сортировки. Помимо ключа, запись может содержать дополнительную «сопутствующую информацию», которая не влияет на сортировку, но всегда остается в этой записи.

Отношение порядка « < » на множестве ключей т.о., чтобы для всех 3-х значений ключей а, в, с выполнялись следующие условия:

1. справедливо одно и только одно из соотношений: а< b, a=b, b<a (закон Трихотомии);
2. если а< b и b<с, то а<с (закон Транзит-ти).

Эти 2 свойства определяют математическое понятие линейного упорядочения, которое называют еще совершенным упорядочением. Любое множество с отношением « < », удовлетворяемых свойствам 1) и 2), поддается сортировке большинством методом сортировки, который мы будем рассматривать, хотя некоторые из них годятся только для числовых и буквенных ключей с обычным отношением порядка.

Задача сортировки - найти такую перестановку записей p(1), p(2), p(n) после К ключи расположились бы в неубывающем порядке:

Kp(1)<Kp(2)<...<Kp(n)

Сортировка называется устойчивой, если она удовлетворяет дополнительному условию, что записи с одинаковыми ключами остаются в прежнего порядке, то есть:

Р(i)<Р(j), если Kp(i)=Kp(j) и i<j

Обычно сортировку подразделяют на 2 класса:

* внутреннюю, если записи, которые сортируются, находятся в операционной памяти;
* внешнюю, если некоторые из записей, которые сортируются, находятся во вспомогательной памяти.

Ограничим наше рассмотрение лишь внутренними сортировками. Алгоритмы сортировки можно подразделить на несколько групп:

A. Сортировка вставками. Элементы просматриваются по одному, и каждый новый  
элемент вставляется в подходящее место среди ранее упорядоченных элементов.

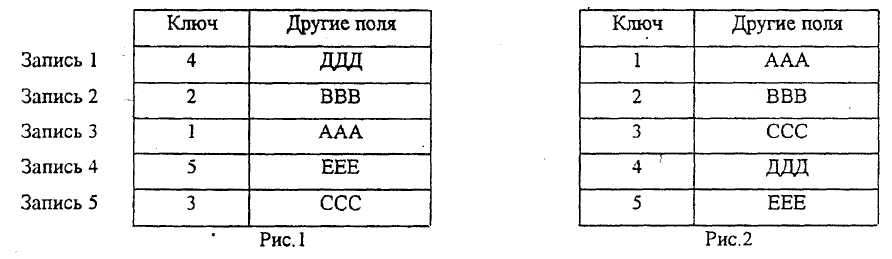
B. Обменная сортировка. Если два элемента расположены не по порядку, то они  
меняются местами. Этот процесс повторяется до тех пор, пока элементы не будут упорядочены.

C. Сортировка посредством выбора. Сначала выделяется наименьший (или, может быть,  
наибольший) элемент и каким-либо- образом отделяется от остальных, затем выбирается  
наименьший (наибольший) из оставшихся и так далее.

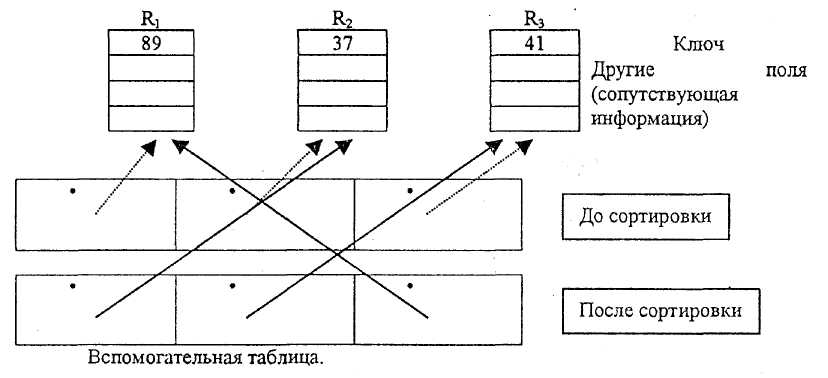
Д. Сортировка подсчетом. Каждый элемент сравнивается со всеми остальными, окончательное положение элемента определяется подсчетом числа наименьших ключей.

Е. Специальная сортировка, которая хороша для нескольких элементов, но не поддается простому обобщению на случай большего числа элементов.

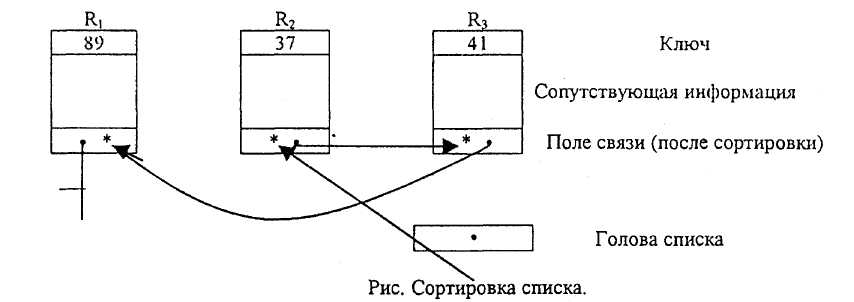
Сортировка выполняется или над самими записями, или над указателями некоторой вспомогательной таблицы, например, рассматриваем рис.1, на котором представлен файл с 5 записями. Если этот файл отсортировать в возрастающем порядке по указанному числовому ключу, то результирующий файл будет таким, как показано на рис.2. В этом случае были отсортированы и сами записи.



Если записи и/или ключи занимают несколько слов памяти, то часто лучше составить новую таблицу адресов (ссылок), которые указывают на записи и работают с этими адресами, не перемещая громоздкие записи. Такой метод называется сортировкой таблицы адресов.



Если ключи короткие, а сопутствующая информация в записях велика, то для повышения скорости ключи можно вынести в таблицу адресов; это называется сортировкой ключей. Другие схемы сортировки используют вспомогательное поле связи, которое включается в каждую запись. Связи обрабатываются т.о., что в результате все записи оказываются связанными в линейный список, в котором каждая связь указывает на следующую по порядку запись. Это называется сортировкой списка.



Достаточно хороший алгоритм затрачивает на сортировку N записей время порядка NlogN; при этом требуется около log N «проходов» по данным.

При N--»oo время работает на N (log)2 , если все ключи различны, так как и размеры ключей увеличиваются с ростом N, но практически N всегда остается ограниченным.

Сортировка вставками.

Простые вставки. Пусть 1 < j < N и записи R1,..., Rj-1 уже размещены так, что К1 < К2 < ... Kj1

Будем сравнивать по очереди Kj с Kj-1 Kj-2> ... до тех пор, пока не обнаружим, что запись Rj следует вставить в R1 и R1+i; тогда подвинем записи Ri+1, ..., Rj-1 на одно место вверх и поместим новую запись в позицию i+1. Удобно совмещать операции сравнения и перемещения, перемежая их друг с другом. Поскольку запись Rj как бы «проникает на положенный ей уровень», этот способ часто называют просеиванием или погружением.

Алгоритм (Сортировка простыми вставками). Записи R1...,RN переразмещаются на том же месте; после завершения сортировки их ключи будут упорядочены: К1 < ... < KN.

Шаг 1 [Цикл по j]. Выполнить шаги 2-5 при j=2,3,...,N и после этого завершить алгоритм.

Шаг 2 [Установить i, К, R]. Установить i=j-l, K=Kj, R=Rj. (В последующих шагах мы попытаемся вставить запись R в нужное место, сравнивая К с Кi при убывающих значениях i).

Шаг 3 Сравнить К, K1. Если К > Ki, то перейти к шагу 5 (Мы нашли искомое место для записи R).

Сортировка простыми вставками.

98 12 13 7 3 21 99 128

j=2 K2=

i=l K=12 R=R2

R2=R1 i=0

12 98 13 7 3 21 99 128

j=3

i=2 K=K3=13 R=R3 R3=R2 i=2

12 13 98 7 3 2199 128

j=4

i=3 K=K4=7 R=R4 R4=R3 i=3

12 13 7 98

j=2

i=l K=K2=13 R=R2

12 13 7 98

j=3

i=2 K=K3=7 R=R3 R3=R2 i=2

12 7 13 98

j=4

i=3 K=K4=98 R=R4

12 7 13 98

j=2

i=l K=K2=7 R=R2 R2=R1 i=0

7 12 13 98

j=3

i=2 K=K3=13 R=R3

7 12 13 98

j=4

i=3 K=K4=98 R=R4

7 12 13 98

10 просмотров.

Шаг 4 [Переместить Ri, уменьшить i]. Установить Ri+1=Ri, i =i-l. Если i>0, то вернуться к шагу 3. (Если i=0, то К - наименьший из рассмотренных до сих пор ключей, а значит, запись R должна занять первую позицию).

Шаг 5 [R на место Ri+1]. Установить Rj+1= R

**2. Сортировка Бетчера**

Параллельная сортировка Бэтчера. Если мы хотим получить, алгоритм Обм. С, время работы которого имеет порядок, меньший N2, то нужно подбирать для сравнений некоторые пары несоседних ключей (Кi, Kj).

Схема сортировки Бетчера несколько напоминает сортировку Шелла, но сравнения выполняются по новому, так что распространение обменов не обязятельно. Метод Бэтчера называют еще «Обм. С со слиянием», так как в нем происходит слияние пар отсортированных последовательностей.

Алгоритм (Бэтчера). Записи R1 ..., Rn переразмещаются на том же месте; после завершения сортировки их ключи будут упорядочены K1 < ... < KN. Предполагается, что N>2.

Шаг 1 [Начальная установка р]. Установить р=2t-1 где t=[log2N] - наименьшее целое число такое, что 2t>N. Шаги 2-5 будут выполняться с р=2t-1 ,2t-1, ... 1.

Шаг 2 [Начальная установка q, г, d]. Установить q=2t-1, r=0, d=p.

Шаг 3 [Цикл по i]. Для всех i, таких, что o<i<N-d и iAp=r выполнить шаг 4. Затем перейти к шагу 5. (Здесь через iAp обозначена операция "логические и" над двоичными представлениями чисел i и р; все биты результата равны 0, кроме тех, для которых в соответствующих позициях i и р находятся единичные биты.

Так, 13A21=(1101)2A(10101)2=(00101)2=5. К этому моменту d - нечетное кратное р (т.е. частное от деления d на р нечетно), а р- степень двойки, так что i^p^=(i+d) ^p; (отсюда следует, что действия шага 4 можно выполнять при всех нужных значениях i в любом порядке или даже одновременно).

Шаг 4 [Сравнение / обмен Ri+1 и Ri+d+1]. Если Ki+1>Ki+d+1 , то поменять местами Ri+1 и Ri+d+l

Шаг 5 [Цикл по q]. Если q=p, то установить d=q-p, q=q/2, r=p и возвратиться к шагу 3.

Шаг 6 [Цикл по р]. (К этому моменту перестановка К1 К2, ... KN будет р-упорядочена). Установить p=[p/2]. Если р>0, то возвратиться к шагу 2.

Быстрая сортировка Бетчера. В методе Бетчера последовательность сравнений предопределений каждый раз сравнивается одни и те же пары ключей независимо от того, что мы могли узнать о файле из предыдущих сравнений. Это утверждение в большей мере справедливо и применительно к методу пузырька хоть алгоритм Бетчера и использует в ограниченой степени получение сведения, с тем, сократить количество работы в правом конце файла.

**3. Структура, задачи и формализация предметной области**

В основе логического моделирования лежит формальная система, задаваемая четвёркой вида М=<Т, Р, А, В>. Множество Т - множество базовых элементов различной природы. Для этого множества существует некоторый способ определения принадлежности или не принадлежности любого элемента к этому множеству. Процедура проверки может быть любой, но за конечное число шагов она должна дать положительный или отрицательный результат на вопрос, является ли *х* элементом множества Т. Обозначим эту процедуру П(Т).

Множество Р - множество синтаксических правил. С их помощью из элементов Т образуются синтаксически правильные совокупности. Существует процедура П(Р), с помощью которой за конечное число шагов можно получить ответ на вопрос, является ли совокупность X синтаксически правильной.

В множестве синтаксически правильных совокупностей выделяется некоторое подмножество А. Элементы А называются *аксиомами.* Процедура П(А) даёт ответ ка вопрос, принадлежит ли данная синтаксически правильная совокупность к множеству А.

Множество В - множество правил вывода. Применяя их к элементам А, можно получить новые синтакагчески правильные совокупности, к которым можно применять правила из В. Если имеется процедура П(В), с помощью которой можно определить для любой синтаксически правильной совокупности, является ли она выводимой, то соответствующая формальная система называется *разрешимой.*

Все предметы и события, которые составляют основу общего понимания необходимой для решения задачи информации, называются *предметной областью.* Предметная область представляется из реальных или абстрактных объектов, называемых *сущностями.*

Сущности предметной области находятся определённых *отношениях* друг к другу (ассоциациях), которые также можно рассматривать как сущности и включать в предметную область. Между сущностями наблюдаются различные отношения подобия. Совокупность подобных сущностей составляет *класс сущностей,* являющийся новой сущностью предметной области.

Отношения между сущностями выражаются с помощью суждений. *Суждение* - это мысленно возможная ситуация, которая может иметь место для представляемых сущностей или не иметь места.

В языке суждения отвечают *предложения.* Суждения и предложения также можно рассматривать как сущность и включать в предметную область.

Языки, предназначенные для описания предметных областей, называются *языками представления знаний.* Универсальным языком представления знаний является естественный язык. Однако использование естественного языка в системах машинного представления знаний наталкивается на большие трудности, связанные с двусмысленностью, нерегулярностью и т.д. Но главное препятствие заключается в отсутствии формальной семантики естественного языка, которая имела бы достаточно эффективную операционную поддержку.

Для представления математического знания в математической логике давно пользуются *исчислением предикатов,* которое имеет ясную формальную семантику и операционную поддержку (разработаны механизмы вывода).

Описания предметных областей, выполненные в логических языках, называются (формальными) *логическими моделями.*