МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Московский Авиационный Институт
(Национальный Исследовательский Университет)

Институт №8 "Информационные технологий и прикладная математика"
Кафедра 806 "Вычислительная математика и программирование"

Курсовая работа
по курсу "Вычислительные системы"
2 семестр

Задание 9
Сортировка и поиск

Автор работы:
студент 1 курса, группы М8О-106Б-20
Леухин М. В.
Преподаватель:
Дубини А. В.

Москва, 2021

Содержание

1 Постановка задачи

Задание: Составить программу на языке Си с использованием процедур и функций для сортировки таблицы заданным методом и двоичного поиска по ключу в таблице. Программа должна вводить значения элементов неупорядоченной таблицы и проверять работу процедуры сортировки в трёх случаях:

1. Элементы таблицы с самого начала упордочены.
2. Элементы таблицы расставлены в обратном порядке.
3. Элементы таблицы не упорядочены.

Для каждого вызова процедуры сортировки необходимо печатать исходное состояние таблицы и результат сортировки. После выполнения сортировки программа должна вводить ключи и для каждого из них выполнять поиск в упорядоченной таблице с помощью процедуры двоичного поиска и печатать найденные элементы, если они присутствуют в таблице.

Метод сортировки: сортировка Шелла.

Тип ключа: целый (8 байт), данные хранятся вместе с ключём.

2 Основная часть

2.1 Сортировка

Сортировка — процесс перестановки заданного множества объектов в некотором порядке. Цель сортировки — облегчить последующий поиск элементов в таком отсортированном множестве.

Сортировки представляют собой хороший пример задач, для решения которых существует множество различных алгоритмов, каждый из которых имеет свои достоинства и недостатки. Выбор алгоритма зависит от конкретных условий. Телефонные книги, словари, оглавления библиотек, прайс-листы — вот примеры отсортированных для поиска множеств хранимых объектов.

Выбор конкретного алгоритма зависит от структуры обрабатываемых данных. В случае сортировки эта зависимость столь глубока, что соответствующие методы распадаются на два почти непересекающихся класса - *сортировку массивов* и *сортировку файлов (последоватлеьностей)*. Иногда их называют *внутренними* и *внешними* сортировками, поскольку массивы хранятся в основной (оперативной, внутренней) памяти машины с произвольным доступом, а файлы обычно размещаются в медленной, но более ёмкой, дешёвой и долговременной внешнй памяти.

Для постановки задачи сортировки введём некоторые понятия и обозначения. Сортируемые элементы будем обозначать $a_1, a_2, ..., a_n$. Сортировка есть некоторая перестановка этих элементов $a_{k_1}, a_{k_2}, ..., a_{k_n}$ ($k_1, k_2, ..., k_n$ — перестановка последовательности $1, 2, ..., n$), для которой выполнено некоторое отношение порядка f:

$$f(a_{k_1}) \leq f(a_{k_2}) \leq ... \leq f(a_{k_n})$$

В случае, когда элементы $a_1, a_2, ..., a_n$ непосредственно сравнимы, отношение порядка f вычислять не требуется, а достаточно хранить сами сравниваемые компоненты в соответствующих полях элементов a_i. Так же, как и в случае поиск, сравнимые поля элементов сортировки называют *ключами*. Типы других компонент элементов, хранимых с ключами, могут быть самыми разнообразными, и поэтому элементы обычно имеют комбинированный тип.

Метод сортировки называется *устойчивым*, если в процессе сортировки относительное расположение элементов с равными ключами не изменяется. Устойчивость сортировки полезна, когда предполагается последующее упорядочивание по вторичным ключам среди равнозначных элементов.

2.2 Сортировка вставками

По задания необходимо использовать сортировку Шелла, идея которой основана на сортировке вставками, поэтому целесообразно будет для начала рассмотреть сортировку вставками.

Сортировка вставками — алгоритм сортировки, в котором элементы входной последовательности просматриваются по одному, и каждый новый элемент поступивший элемент размещается в подходящее место среди ранее упорядоченных элементов.

Этот метод широко используется при игре в карты: игрок обычно располагает имеющиеся карты в левой руке в порядке возрастания ранга, а сдаваемая карта вставляется сообразно её рангу после последней карты того же ранга. Сортировка вставкой как раз

представляет собой вариант этой карточной идеи. Сортируемые элементы мысленно делятся на уже готовую (отсортированную) последовательность $a_1, ..., a_{i-1}$ и исходную (неупорядоченную) последовательность $a_i, ..., a_n$. В начальный момент времени первая последовательность должна быть пуста, но мы включим в неё первый элемент множества, так как последовательность из одного элемента всегда упорядочена. Этот элемент необязательно будет минимальным, и впоследствии он может быть переставлен. Ниже приведён пример работы алгоритма прямой вставки для последовательности 44 55 12 42 94 18 06 67 (вставленные элементы выделены полужирным шрифтом, вставляемые — курсивом):

Исходные ключи	44 *55* 12 42 94 18 06 67
$i = 2$	44 **55** *12* 42 94 18 06 67
$i = 3$	**12** 44 55 *42* 94 18 06 67
$i = 4$	12 **42** 44 55 *94* 18 06 67
$i = 5$	12 42 44 55 **94** *18* 06 67
$i = 6$	12 **18** 42 44 45 94 *06* 67
$i = 7$	5 **06** 12 18 42 44 45 94 *67*
$i = 8$	5 06 12 18 42 44 45 **67** 94

Вычислительная сложность сортировки вставками — $O(n^2)$.

2.2.1 Псевдокод сортировки вставками

На вход процедуре сортировки подаётся массив $A[1...n]$, состоящий из элементов последовательности $A[1], A[2], ..., A[n]$, которые требуется отсортировать. Псевдокод алгоритма:

Algorithm 1 Сортировка вставками

```
1: for i = 2 to n do
2:     x = A[i]
3:     j = i
4:     while (j > 1 and A[j − 1] > x) do
5:         A[j] = A[j − 1]
6:         j = j − 1
7:     end while
8:     A[j] = x
9: end for
```

2.3 Сортировка Шелла

Сортировка Шелла — алгоритм сортировки, являющийся усовершенствованным вариантом сортировки вставками. Идея этого метода состоит в сравнении элементов, стоящих не только рядом, но и на определённом расстоянии друг от друга.

Идея была предложена в 1959 г. Д. Шеллом. Для ускорения он предложил выделять в сортируемой последовательности периодические подпоследовательности регулярного шага, в каждой из которых отдельно выполняются обычные сортировки сортировки вставками. При сортировке Шелла сначала сравниваются и сортируются между собой значения, стоящие друг от друга на расстоянии d (о выборе d ниже). После этого

процедура повторяется для некоторых меньших значения d, а завершается сортировка Шелла упорядочиванием элементов при $d = 1$, то есть обычной сортировкой вставками. Ниже приведён пример работы сортировки Шелла:

Исходные ключи	44 55 12 42 94 18 06 67
$d = 4$	44 18 06 42 94 55 12 67
$d = 2$	06 18 12 42 44 55 94 67
$d = 1$	06 12 18 42 44 55 67 94

Среднее время работы алгоритма зависит от длин промежутков d, на которых будут находиться сортируемые элементы исходного массива. При кратном уменьшении шага, например, 8 4 2 1 мы в конце концов получим две автононмные последовательности и на завершающем проходе нам придётся "много вставлять". Избежать этого можно по рецепту Д. Кнута, используя последовательность

1 4 13 40 121 364 1093 3280 9841 ... $3n + 1$ ...

Эта последовательность приводит к хорошему перемешиванию организуемых подпоследовательностей и сводит к минимуму их изолированность, и на последней стадии сериализации вставка будет идти почти без ресурсоёмких сдвигов. Вот ещё несколько вариантов:

- Первоначально используемая Шеллом последовательность длин промежутков: $d_1 = n/2, d_i = d_{i-1}/2, d_k = 1$. В худшем случае, сложность алгоритма составит $O(n^2)$
- Предложенна Хиббардом последовательность: все значения $2^i - 1 \leq n, i \in \mathbb{N}$. Такая последовательность шагов приводит к сложности $O(n^{3/2})$.
- Предложенная Праттом последовательность: все значения $2^i \times 3^j \leq n/2, i, j \in \mathbb{N}$. В таком случае сложность алгоритма составляет $O(n(log\ n)^2)$.

Пример реализации сортировки Шелла на языке Си:

```
1  void shell_sort(int *array, int size){
2      for(int s=size/2; s>0; s/=2){
3          for(int i=0; i<size; i++){
4              for(int j=i+s; j<size; j+=s){
5                  if(array[i] > array[j]){
6                      int temp = array[j];
7                      array[j] = array[i];
8                      array[i] = temp;
9                  }
10             }
11         }
12     }
13 }
```

3 Вывод

Алгоритмы сортировки играют большую роль в информатике и жизни в целом. Они не просто значительно упрощают, а иногда в принципе делают возможным поиск элементов в определённых последовательностях. Представьте себе поиск номера в телефонной книге, если бы номера в ней не были отсортированы, или поиск нужного вам слова среди 280 тысяч других в словаре В. И. Даля.

Существует огромное количество алгоритмов сортировки, среди которых нельзя однозначно выбрать лучшие (но можно выбрать худшие, например Bogosort со средним временем работы $O(n \times n!)$), ведь различные алгоритмы сортировки по-разному себя показывают на рвзных наборах данных. Например, есть быстрая сортировка Хоара, которая в среднем работает за $O(n \times log\ n)$ и во многих случаях быстрее, чем рассмотренная сортировка Шелла, однако легко деградирует до $O(n^2)$, что ещё хуже, чем худшее гарантированное время сортировки Шелла.

4 Список источников

- Методические указания к выполнению курсовых работ. Зайцев В. Е.
- Курс информатики. Гайсарян С. С., Зайцев В. Е.