

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего об­разования

**"МИРЭА - Российский технологический университет"**

# РТУ МИРЭА

Институт информационных технологий (ИТ)

Кафедра МОСИТ

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ**

**по дисциплине**

**«**СТРУКТУРЫ И АОД**»**

Тема: Карманная сортировка и сортировка подсчетом

|  |  |  |  |
| --- | --- | --- | --- |
| Выполнил студент группы ИКБО-16-19 | |  | Беляевский Н.А. |
| Принял |  |  | Кораблин Ю.П. | |
| Практические работы выполнены | «\_\_»\_\_\_\_\_\_\_2020 г. |  |  |
|  |  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_2020 г. |  |  |
|  | |

Москва 2020

**СОДЕРЖАНИЕ**

1. Постановка задачи3

2. Сортировка подсчетом3

3. Карманная сортировка4

4. Время выполнения сортировок5

5. Выводы5

1. **Постановка задачи**

Реализовать алгоритмы карманной сортировки и сортировки подсчетом.

Измерить время выполнения сортировок при разных значениях n – размера сортируемого массива.

1. **Сортировка подсчетом**

**Сортировка подсчётом**— алгоритм сортировки, в котором используется диапазон чисел сортируемого массива (списка) для подсчёта совпадающих элементов.

Иногда сортируемые элементы имеют небольшой диапазон возможных значений. Например, если необходимо отсортировать список натуральных чисел, каждое из которых не более трех-пяти разрядов в длину. Однако количество сортируемых чисел настолько велико, что нужно обработать список за линейное время.

Исходная последовательность чисел длины n, а в конце отсортированная, хранится в массиве A. Также используется вспомогательный массив C с индексами от 0 до k-1 изначально заполняемый нулями.

void Counting\_Sort(int a[], int b[], int k) {

double start, end;

start = clock();

int\* C = new int[k];

for (int i = 0; i < k; i++)

C[i] = 0;

for (int j = 0; j < 1000000; j++) {

C[a[j]] = C[a[j]] + 1;

}

for (int i = 1; i < k; i++) {

C[i] = C[i] + C[i - 1];

}

for (int j = 1000000 - 1; j > -1; j--) {

b[C[a[j]] - 1] = a[j];

C[a[j]] = C[a[j]] - 1;

}

end = clock();

cout << (end - start) / 1000.0;

}

1. **Карманная сортировка**

**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.

void bucketSort(double arr[], int n)

{

double start, end;

start = clock();

vector < vector <double> > b(n, vector <double>());;

for (int i = 0; i < n; i++)

{

b[n \* arr[i]].push\_back(arr[i]); // Распределение по корзинам

}

for (int i = 0; i < n; i++) { // Сортировка корзин

double temp, item;

for (int j = 1; j < b[i].size(); i++)

{

temp = b[i][j];

item = j - 1;

while (item >= 0 && b[i][item] > temp)

{

b[i][item + 1] = b[i][item];

item--;

}

b[i][item + 1] = temp;

}

}

int k = 0;

for (int i = 0; i < n; i++) // Слияние отсортированных корзин

{

for (int j = 0; j < b[i].size(); j++)

arr[k++] = b[i][j];

}

end = clock();

cout << (end - start) / 1000.0 << endl;

}

1. **Время выполнения сортировок**

|  |  |  |  |
| --- | --- | --- | --- |
| Размерность массива | 10000 | 100000 | 1000000 |
| Карманная сортировка | 0.013 сек | 0.112 сек | 0.498 сек |
| Сортировка подсчетом | 0.001 сек | 0.004 сек | 0.023 сек |

1. **ВЫВОДЫ**

В ходе практической работы, реализовал алгоритмы сортировки подсчетом и карманной сортировки на языке программирования С++, измерил время работы каждой сортировки с разными значениями n (размерность массива).