МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждения образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологии

Кафедра Информационных систем и технологий

Специальность 1-40 05 01 Информационные системы и технологии

**РЕФЕРАТ**

на тему:

«Сортировка расчёской и Stooge sort»

Выполнил студент Iкурса 3 группы специальности ИСиТ Задорожный Н.П

(Ф.И.О)

Руководитель

(учен. степень, звание, должность, подпись, Ф.И.О.)

Минск 2019

Stooge sort - рекурсивный алгоритм сортировки с временной сложностью O(n^2.71). Минусы: используется слишком много дополнительной памяти из-за рекурсии.

{\displaystyle O(n^{\log \_{1{,}5}{3}})\approx O(n^{2.71})}

void stoogesort(int \*m, int left, int right) // Stooge sort

{

int tmp, k;

if (m[left] > m[right]) // Сравнение на концах и поменять местами в случае если это верности этого неравенства

{

tmp = m[left];

m[left] = m[right];

m[right] = tmp;

}

if ((left + 1) >= right) // Если массив меньше чем из 2 элементов закончить сортировку

return;

k = (int)((right - left + 1) / 3);

stoogesort(m, left, right - k); //Рекурсивно вызвать Stooge sort для первых 2/3 списка

stoogesort(m, left + k, right); //Рекурсивно вызвать Stooge sort для последних 2/3 списка

stoogesort(m, left, right - k); //Рекурсивно вызвать Stooge sort для первых 2/3 списка снова

}

Сортировка расчёской - Модификация [пузырьковой сортировки](http://algolab.valemak.com/bubble). Нотация Big O для сортировки расчёской O(n\*log(n)) в лучшем случае. Сортировка расческой оказывается намного быстрее, чем пузырек/шейкер, она даже может обогнать «быструю» сортировку. Но есть и минус — она не устойчивая.

void CombSort(int \*m, int n) // Сортировка расчёской

{

int tmp, k;

int s = n; // готовим начальный шаг

while (n > 1) {

s /= 1.247f; //формируем шаг на этом проходе с помощью коэффициента, полученного с помощью золотого сечения.

if (s < 1) s = 1; // 0 быть не может, присвоим 1

k = 0; // нет перестановок

for (int i = 0; i + s < n; ++i) { // двигаемся, пока сравниваемый элемент (на s от текущего) в массиве

if (m[i] > m[i + s]) {

tmp = m[i];

m[i] = m[i + s];

m[i + s] = tmp;

k = i;

}

}

if (s == 1) // шаг равен 1, как в обычном пузырьке.

n = k + 1;

}

}

Функция main :

void main()

{

setlocale(LC\_ALL, "Russian");

int arr[10]; // Создадим массив из 10 целочисленных элементов

for (int i = 0; i < 10; i++) { // Заполнение массива случайными числами

arr[i] = rand() % 20;

cout << arr[i] << ' '; // Вывод элементов до сортировки

}

int Key;

cout << "\n1- Сортировка расчёской | 2 - Stooge Sort\n";

cin >> Key;

switch (Key) {

case 1:

cout << endl << "Сортировка расчёской";

CombSort(arr, 10);

break;

case 2:

cout << endl << "Stooge sort";

stoogesort(arr, 0,9); // Вызов функции сортировки

break;

default:

cout << "Некорректный выбор";

system("pause");

return;

break;

}

cout << endl;

for (int i = 0; i < 10; i++) { // Вывод отсортированного массива

cout << arr[i] << ' ';

}

system("pause");

}

Результаты выполнения:



