МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧЕРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ОТЧЕТ

по лабораторной работе №2

«Алгоритмы сортировки»

Выполнил: Лесько М.И.

Группа: ПО-11

Проверил: Глущенко Т.А.

Брест 2023

***Цель работы***: Реализовать 3 алгоритма сортировки и провести экспериментальное исследование их эффективности.

Код сортировки подсчетом:

#include <iostream>

#include <time.h>

using namespace std;

void countSort(int arr[], int n) {

int arr1[10];

int count\_arr[10];

int x = arr[0];

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

if (arr[i] > x)

x = arr[i];

}

for (int i = 0; i <= x; ++i) {

count\_arr[i] = 0;

}

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

count\_arr[arr[i]]++;

}

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

count\_arr[i] += count\_arr[i - 1];

}

for (int i = n - 1; i >= 0; i--) {

arr1[count\_arr[arr[i]] - 1] = arr[i];

count\_arr[arr[i]]--;

}

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

arr[i] = arr1[i];

}

}

void display(int arr[], int n) {

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

cout << arr[i] << " ";

cout << endl;

}

int main() {

setlocale(LC\_ALL, "ru");

double time\_spent = 0.0;

clock\_t begin = clock();

int arr[10] = { 4, 2, 5, 8, 3, 3, 1, 9, 0, 6 };

int n = sizeof(arr) / sizeof(arr[0]);

countSort(arr, n);

display(arr, n);

clock\_t end = clock();

time\_spent += (double)(end - begin) / CLOCKS\_PER\_SEC;

printf("Время выполнения %f секунд", time\_spent);

return 0;

}

***Проведём подсчёт с разной размерностью массива***

**Массив: 10:**



**Массив: 20:**



**Массив: 30:**



**Массив: 40:**

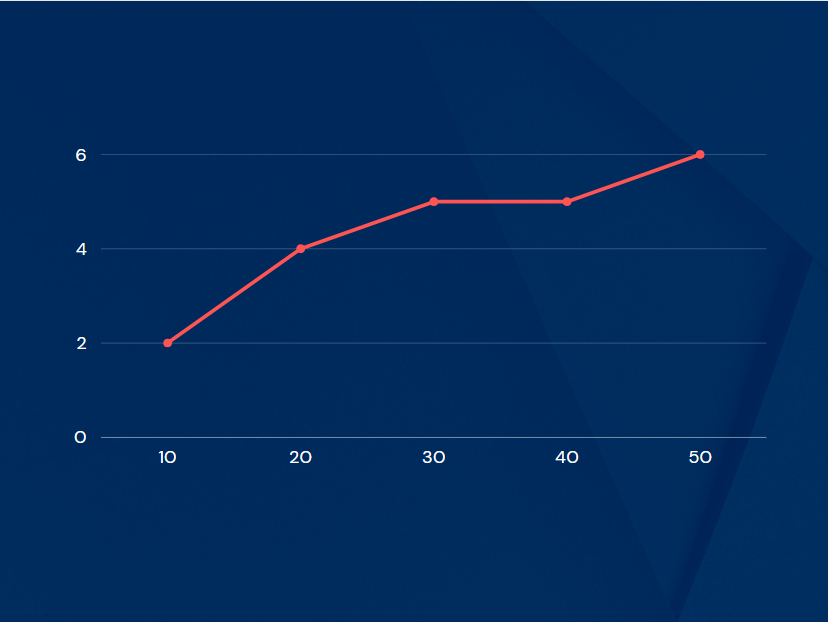


**Массив: 50:**



Сложность О(n+k)

***Изобразим прогрессию времён от размерности массива в графике***

******

Код пузырьковой сортировки:

#include <iostream>

#include <time.h>

using namespace std;

int main()

{

setlocale(LC\_ALL, "ru");

double time\_spent = 0.0;

clock\_t begin = clock();

const int n = 10;

int arr[n];

int i = 0;

cout << "Массив " << endl;

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

arr[i] = 0 + rand() % 100000;

int temp;

for (int i = 0; i < n - 1; i++) {

for (int j = 0; j < n - i - 1; j++) {

if (arr[j] > arr[j + 1]) {

temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

}

}

}

cout << "Отсортированный массив " << endl;

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

cout << arr[i] << endl;

}

cout << endl;

clock\_t end = clock();

time\_spent += (double)(end - begin) / CLOCKS\_PER\_SEC;

printf("Время выполнения %f секунд", time\_spent);

return 0;

}

***Проведём подсчёт с разной размерностью массива***

**Массив: 10:**



**Массив: 20:**



**Массив: 30:**



**Массив: 40:**

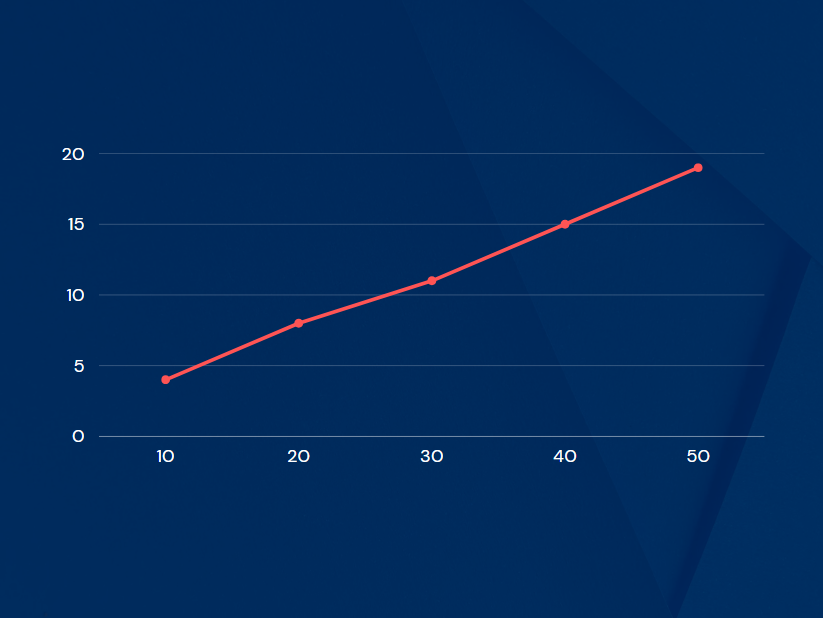


**Массив: 50:**



Сложность О(n^2)

***Изобразим прогрессию времён от размерности массива в графике***



Код сортировки слиянием:

#include <iostream>

#include <vector>

#include <time.h>

using namespace std;

void merge(vector<int>& arr, int low, int mid, int high)

{

int n1 = mid - low + 1;

int n2 = high - mid;

vector<int> leftArr(n1), rightArr(n2);

for (int i = 0; i < n1; i++)

leftArr[i] = arr[low + i];

for (int j = 0; j < n2; j++)

rightArr[j] = arr[mid + 1 + j];

int i = 0;

int j = 0;

int k = low;

while (i < n1 && j < n2) {

if (leftArr[i] <= rightArr[j]) {

arr[k] = leftArr[i];

i++;

}

else {

arr[k] = rightArr[j];

j++;

}

k++;

}

while (i < n1) {

arr[k] = leftArr[i];

i++;

k++;

}

while (j < n2) {

arr[k] = rightArr[j];

j++;

k++;

}

}

void mergeSort(vector<int>& arr, int low, int high)

{

if (low < high) {

int mid = low + (high - low) / 2;

mergeSort(arr, low, mid);

mergeSort(arr, mid + 1, high);

merge(arr, low, mid, high);

}

}

int main()

{

setlocale(LC\_ALL, "ru");

double time\_spent = 0.0;

clock\_t begin = clock();

vector<int> arr = { 14, 33, 27, 10, 35, 19, 42, 44, 7, 2 };

int n = arr.size();

mergeSort(arr, 0, n - 1);

cout << "Отсортированный массив: ";

for (auto i : arr)

cout << i << " ";

cout << endl;

clock\_t end = clock();

time\_spent += (double)(end - begin) / CLOCKS\_PER\_SEC;

printf("Время выполнения %f секунд", time\_spent);

return 0;

}

**Массив: 10:**



**Массив: 20:**



**Массив: 30:**



**Массив: 40:**

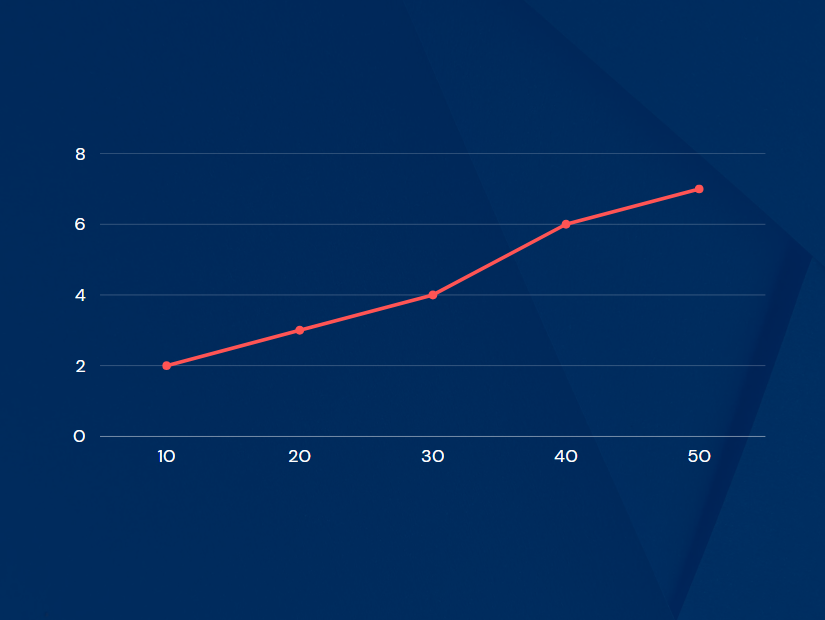


**Массив: 50:**



Сложность О(n log n)

***Изобразим прогрессию времён от размерности массива в графике***



Вывод: Изучил и применил на практике различные виды сортировок.