КПІ ім. Ігоря Сікорського

Інститут прикладного системного аналізу

Кафедра Системного проектування

Алгоритми та структури даних

**Лабораторна робота №3**

**«Дослідження елементарних алгоритмів сортування та пошуку»**

Виконав:

Студент групи ДА-02

Рудік Андрій

Варіант №16

Мета роботи:

Ознайомитись і дослідити елементарні алгоритми сортування: “бульбашкою”, вибором, вставками; алгоритм перетасовки Фішера-Йєтса та алгоритми пошуку: лінійний пошук, бінарний пошук. Набути навичок їх реалізації мовою програмування С/C++ та порівняти ці алгоритми.

Варіант №16.

Завдання №1

Базове сортування “бульбашкою”, пошук елементів у масиві

1. Створити масив на N = 1000 елементів та заповнити його випадковими цілими числами від -2000 до 2000
2. Написати функцію для сортування алгоритмом “бульбашкою”, відсортувати масив за зростанням
3. Написати функцію для пошуку елементу в масиві лінійним алгоритмом, яка повертає індекс знайденого елементу або -1, якщо такий елемент не знайдено
4. Написати функцію для пошуку елементу в масиві бінарним алгоритмом, яка повертає індекс знайденого елементу або -1, якщо такий елемент не знайдено
5. Порівняти швидкодію алгоритмів пошуку (пошук у масиві випадкових чисел від -2000 до 2000)

Результат



Лістинг

#include <iostream>

#include <clocale>

#include <ctime>

#include <cstdlib>

#include <cmath>

#define NUMBER\_OF\_TESTS 100000

using *namespace* std;

*void* fillWithRandom(*int* \**arr*, *int* *length*) {

  for (*int* i = 0; i < *length*; i++) {

*arr*[i] = rand() % 4001 - 2000;

  }

}

*void* printArr(*int* \**arr*, *int* *length*) {

  for (*int* i = 0; i < *length*; i++) {

    cout << *arr*[i] << " ";

  }

  cout << endl;

}

*void* swapElements(*int* \**arr*, *int* *firstPos*, *int* *secondPos*) {

*int* tmp = *arr*[*firstPos*];

*arr*[*firstPos*] = *arr*[*secondPos*];

*arr*[*secondPos*] = tmp;

}

*void* bubbleSort(*int* \**arr*, *int* *length*, *bool* (\**compare*)(*int*, *int*)) {

*bool* sorted;

*int* counter = 1;

  do {

    sorted = true;

    for (*int* i = 0; i < *length* - counter; i++) {

      if (*compare*(*arr*[i], *arr*[i + 1])) continue;

      sorted = false;

      swapElements(*arr*, i, i + 1);

    }

    counter++;

  } while (!sorted && *length* - counter > 0);

}

*int* findLinear(*int* \**arr*, *int* *length*, *int* *element*) {

*int* index = -1;

  for (*int* i = 0; i < *length*; i++) {

    if (*arr*[i] == *element*) {

      index = i;

      break;

    }

  }

  return index;

}

*int* findBinary(*int* \**arr*, *int* *length*, *int* *element*) {

*int* start = 0;

*int* end = *length*;

*int* mid;

  while (start < end) {

    mid = (start + end) / 2;

    if (*arr*[mid] < *element*) {

      start = mid + 1;

    } else {

      end = mid;

    }

  }

  return *arr*[start] == *element* ? start : -1;

}

*int* main() {

  setlocale(LC\_ALL, "rus");

  srand(time(NULL));

*int* length;

  cout << "Введите длину массива: ";

  cin >> length;

  if (length < 1) {

    cout << "Некорректная длина массива";

    return 1;

  }

*int* \*arr = new *int*[length];

  fillWithRandom(arr, length);

  cout << "Массив случайных чисел от -2000 до 2000 длины " << length << endl;

  printArr(arr, length);

  clock\_t start = clock();

  bubbleSort(arr, length, [](*int* *a*, *int* *b*) { return *a* < *b*; });

*float* bubbleSortTime = (*float*)(clock() - start) / CLOCKS\_PER\_SEC;

  cout << "Отсортированый массив" << endl << "На его сортировку было потрачено " << bubbleSortTime << " секунд" << endl;

  printArr(arr, length);

*int* element;

  cout << "Введите элемент, индекс которого вы желаете узнать: ";

  cin >> element;

*int* linearIndex;

  start = clock();

  for (*int* i = 0; i < NUMBER\_OF\_TESTS; i++) {

    linearIndex = findLinear(arr, length, element);

  }

*float* linearTime = (*float*)(clock() - start) / CLOCKS\_PER\_SEC;

*int* binaryIndex;

  start = clock();

  for (*int* i = 0; i < NUMBER\_OF\_TESTS; i++) {

    binaryIndex = findBinary(arr, length, element);

  }

*float* binaryTime = (*float*)(clock() - start) / CLOCKS\_PER\_SEC;

  if (linearIndex != binaryIndex) {

    cout << "Впечатляюще...";

  } else {

    cout

      << "Индекс этого элемента " << linearIndex << endl

      << "Это было проверено двумя способами: линейным и бинарными" << endl

      << "Это заняло " << linearTime << " секунд для линейного алгоритма поиска" << endl

      << "И " << binaryTime << " секунд для бинарного алгоритма поиска соответственно" << endl

      << "Для поиска было проведено " << NUMBER\_OF\_TESTS << " тестов";

  }

  delete[] arr;

  return 0;

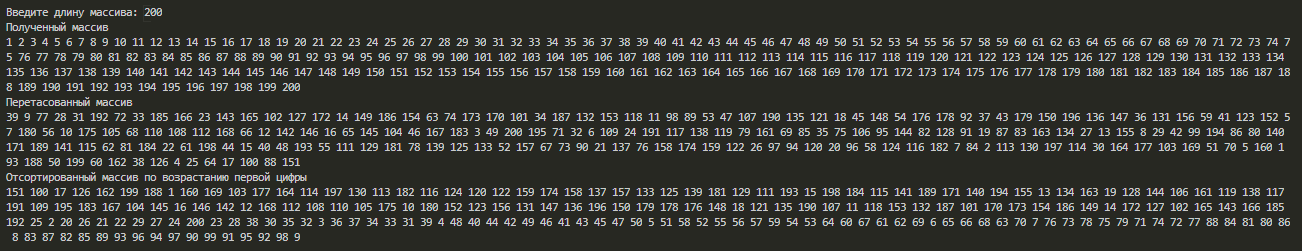
}

Завдання №2

Сортування елементарними алгоритмами по різним критеріям

1. Створити масив на N = 200 елементів та заповнити його послідовними числами від 1 до N
2. Написати функцію для перетасовки масиву алгоритмом Фішера Йєтса, перетасувати масив
3. Сортування вибором за критерієм першої цифри числа у порядку зростання (напр. 148 < 34, адже 1 < 3), при рівності - спочатку менше за значенням.

Результат



Лістинг

#include <iostream>

#include <ctime>

#include <cstdlib>

#include <clocale>

#include <cmath>

using *namespace* std;

*void* printArr(*int* \**arr*, *int* *length*) {

  for (*int* i = 0; i < *length*; i++) {

    cout << *arr*[i] << " ";

  }

  cout << endl;

}

*void* swapElements(*int* \**arr*, *int* *firstPos*, *int* *secondPos*) {

*int* tmp = *arr*[*firstPos*];

*arr*[*firstPos*] = *arr*[*secondPos*];

*arr*[*secondPos*] = tmp;

}

*void* shuffle(*int* \**arr*, *int* *length*) {

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

    swapElements(*arr*, i, rand() % i);

  }

}

*void* selectionSort(*int* \**arr*, *int* *length*, *bool* (\**compare*)(*int*, *int*)) {

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

    for (*int* j = i + 1; j < *length*; j++) {

      if (*compare*(*arr*[i], *arr*[j])) continue;

      swapElements(*arr*, i, j);

    }

  }

}

*void* insertionSort(*int* \**arr*, *int* *length*, *bool* (\**compare*)(*int*, *int*)) {

  for (*int* i = 1; i < *length*; i++) {

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

      if (*compare*(*arr*[j], *arr*[j + 1])) break;

      swapElements(*arr*, j + 1, j);

    }

  }

}

*int* getFirstDigit(*int* *n*) {

  return *n* / round(pow(10, (*int*)log10(*n*)));

}

*bool* compare(*int* *a*, *int* *b*) {

  return getFirstDigit(*a*) < getFirstDigit(*b*);

}

*int* main() {

  srand(time(NULL));

  setlocale(LC\_ALL, "rus");

*int* length;

  cout << "Введите длину массива: ";

  cin >> length;

  if (length < 1) {

    cout << "Некорректная длина массива";

    return 1;

  }

*int* \*arr = new *int*[length];

  for (*int* i = 0; i < length; i++) {

    arr[i] = i + 1;

  }

  cout << "Полученный массив" << endl;

  printArr(arr, length);

  shuffle(arr, length);

  cout << "Перетасованный массив" << endl;

  printArr(arr, length);

  selectionSort(arr, length, compare);

  cout << "Отсортированный массив по возрастанию первой цифры" << endl;

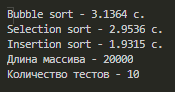
  printArr(arr, length);

  return 0;

}

Порівняння швидкодії

\*Вказані показники включають затрачений час на перетасовку масиву.



Висновок

Я ознайомився і дослідив елементарні алгоритми сортування: “бульбашкою”, вибором, вставками; алгоритм перетасовки Фішера-Йєтса та алгоритми пошуку: лінійний пошук, бінарний пошук. Набув навичок їх реалізації мовою програмування С/C++ та порівняв ці алгоритми.