Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

Лабораторная работа 16

По дисциплине «Основы алгоритмизации и программирования»

На тему «Сортировка массивов»

Выполнила:

Студентка 1 курса 2 группы

Глухова Д.В.

Преподаватель: асс. Андронова М.В.

2023, Минск

5. В соответствии со своим вариантом написать программу сортировок массивов указанными в таблице методами. Исходные массивы заполняются случайными числами. Определить зависимость времени выполнения алгоритмов от количества элементов для каждого из алгоритмов. Выполнить моделирование для массивов различных размеров. Произвести сравнение эффективности алгоритмов (построить график в приложении Excel).

**Вариант 2.**

Ввести массив **А**, в массив **В** перенести все элементы массива **А**, имеющие четный индекс, справа от которых расположены элементы с нечетным значением. Массив **В** отсортировать по убыванию, используя алгоритмы сортировок: сортировка выбором, сортировка Хоара.

|  |
| --- |
| Код программы |
| // Ввести массив А, в массив В перенести все элементы массива А, имеющие четный индекс, справа от которых расположены элементы с нечетным значением.  // Массив В отсортировать по убыванию, используя алгоритмы сортировок : сортировка выбором, сортировка Хоара.  #include <iostream>  using namespace std;  void selectSort(int A[], int size)  {  int k, i, j;  for (i = 0; i < size - 1; i++)  {  for (j = i + 1, k = i; j < size; j++)  if (A[j] > A[k])  k = j;  int c = A[k];  A[k] = A[i];  A[i] = c;  }  }  int getHoarBorder(int A[], int sm, int em)  {  int i = sm - 1, j = em + 1;  int brd = A[sm];  int buf;  while (i < j)  {  while (A[--j] < brd);  while (A[++i] > brd);  if (i < j)  {  buf = A[j];  A[j] = A[i];  A[i] = buf;  };  }  return j;  }  int\* sortHoar(int A[], int sm, int em)  {  if (sm < em)  {  int hb = getHoarBorder(A, sm, em);  sortHoar(A, sm, hb);  sortHoar(A, hb + 1, em);  }  return A;  };  int main() {  setlocale(LC\_ALL, "Rus");  int size, i, j, sizeB = 0, ind = 0, ch = 3;  int A[100], B[100];  cout << "Количество элементов = ";  cin >> size;  cout << "Массив A: \n";  for (i = 0; i < size; i++)  {  A[i] = rand() % 100;  cout << A[i] << " ";  if (i % 10 == 0 && i > 0) cout << "\n";  }  for (i = 0; i < size; i++)  {  j = i + 1;  if (i % 2 == 0 && A[j] % 2 != 0) {  B[ind] = A[i];  sizeB++;  ind++;  }  }  cout << "\nМассив B: ";  for (int i = 0; i < sizeB; i++)  {  cout << B[i] << " ";  }  cout << "\nКак сортировать? 1 - выбором, 2 - Хоар\n";  cin >> ch;  switch (ch) {  case 1:selectSort(B, sizeB); break;  case 2: sortHoar(B, 0, sizeB - 1); break;  }  cout << "\nОтсортированный массив B: ";  for (int i = 0; i < sizeB; i++)  {  cout << B[i] << " ";  }  } |
| Результат в консоли |
|  |

Дополнительные задания.

Вариант 5.

Ввести массивы А и В. В массив С перенести элементы массива А с четным значением и элементы массива В с нечетным значением. Массив С отсортировать по возрастанию, используя алгоритмы сортировок: «пузырек», сортировка выбором.

|  |
| --- |
| Код программы |
| // Ввести массивы А и В. В массив С перенести элементы массива А с четным значением и элементы массива В с нечетным значением.  // Массив С отсортировать по возрастанию, используя алгоритмы сортировок: «пузырек», сортировка выбором.  #include <iostream>  using namespace std;  void selectSort(int A[], int size)  {  int k, i, j;  for (i = 0; i < size - 1; i++)  {  for (j = i + 1, k = i; j < size; j++)  if (A[j] < A[k])  k = j;  int c = A[k];  A[k] = A[i];  A[i] = c;  }  }  void bubbleSort(int a[], int n)  {  int i, j, t;  for (i = 1; i < n; i++)  for (j = n - 1; j >= i; j--)  if (a[j - 1] > a[j])  {  t = a[j - 1];  a[j - 1] = a[j];  a[j] = t;  }  }  int main() {  setlocale(LC\_ALL, "Rus");  int size, i, j, sizeC = 0, ind = 0, ch = 3;  int A[100], B[100], C[100];  cout << "Количество элементов = ";  cin >> size;  for (i = 0; i < size; i++)  {  A[i] = rand() % 20;  if (A[i] % 2 == 0) {  C[ind] = A[i];  ind++;  sizeC++;  }  B[i] = rand() % 40;  if (B[i] % 2 != 0)  {  C[ind] = B[i];  ind++;  sizeC++;  }  }  cout << endl << endl;  cout << "Массив A: ";  for (i = 0; i < size; i++) cout << A[i] << " ";  cout << "\nМассив B: ";  for (i = 0; i < size; i++) cout << B[i] << " ";  cout << "\nМассив C: ";  for (i = 0; i < sizeC; i++) cout << C[i] << " ";  cout << "\nКак сортировать? 1 - выбором, 2 - пузырьком\n";  cin >> ch;  switch (ch) {  case 1:selectSort(C, sizeC); break;  case 2: bubbleSort(C, sizeC); break;  }  cout << "\nОтсортированный массив C: ";  for (i = 0; i < sizeC; i++) cout << C[i] << " ";  } |
| Результат в консоли |
|  |

Вариант 10

Ввести массивы А и В.В массив С перенести те элементы массива А, которые больше максимального элемента массива В. Массив С отсортировать по убыванию, используя алгоритмы сортировок : «пузырек», в сортировка простой вставкой.

|  |
| --- |
| Код программы |
| // Ввести массивы А и В.В массив С перенести те элементы массива А, которые больше максимального элемента массива В.  // Массив С отсортировать по убыванию, используя алгоритмы сортировок : «пузырек», в сортировка простой вставкой.  #include <iostream>  using namespace std;  void bubbleSort(int a[], int n)  {  int i, j, t;  for (i = 1; i < n; i++)  for (j = n - 1; j >= i; j--)  if (a[j - 1] < a[j])  {  t = a[j - 1];  a[j - 1] = a[j];  a[j] = t;  }  }  void insertSort(int\* A, int size)  {  int t, i, j;  for (i = 1; i < size; i++)  {  t = A[i];  for (j = i - 1; j >= 0 && A[j] < t; j--)  A[j + 1] = A[j];  A[j + 1] = t;  }  }  int main() {  setlocale(LC\_ALL, "Rus");  int size, i, j, sizeC = 0, ind = 0, max = 0, ch = 3;  int A[100], B[100], C[100];  cout << "Количество элементов = ";  cin >> size;  for (i = 0; i < size; i++)  {  A[i] = rand() % 20;  B[i] = rand() % 10;  if (B[i] > max) max = B[i];  }  for (i = 0; i < size; i++)  {  if (A[i] > max) {  C[ind] = A[i];  ind++;  sizeC++;  }  }  cout << "Массив A: ";  for (i = 0; i < size; i++) cout << A[i] << " ";  cout << "\nМассив B: ";  for (i = 0; i < size; i++) cout << B[i] << " ";  cout << "\nМаксимальный в B: " << max;  cout << "\nМассив C: ";  for (i = 0; i < sizeC; i++) cout << C[i] << " ";  cout << "\nКак сортировать? 1 - пузырьком, 2 - вставками\n";  cin >> ch;  switch (ch) {  case 1: bubbleSort(C, sizeC); break;  case 2: insertSort(C, sizeC); break;  }  cout << "\nОтсортированный массив C: ";  for (i = 0; i < sizeC; i++) cout << C[i] << " ";  } |
| Результат в консоли |
|  |

Вариант 11

Ввести массивы А и В. В массив С перенести элементы массива А с нечетным значением и элементы массива В с четным значением. Массив С отсортировать по возрастанию, используя алгоритмы сортировок: сортировка Шелла, сортировка Хоара.

|  |
| --- |
| Код программы |
| // Ввести массивы А и В. В массив С перенести элементы массива А с нечетным значением и элементы массива В с четным значением.  // Массив С отсортировать по возрастанию, используя алгоритмы сортировок: сортировка Шелла, сортировка Хоара.  #include <iostream>  using namespace std;  void ShellSort(int mass[], int n)  {  int i, j, step;  int tmp;  for (step = n / 2; step > 0; step /= 2)  for (i = step; i < n; i++)  {  tmp = mass[i];  for (j = i; j >= step; j -= step)  {  if (tmp < mass[j - step])  mass[j] = mass[j - step];  else  break;  }  mass[j] = tmp;  }  }  int getHoarBorder(int A[], int sm, int em)  {  int i = sm - 1, j = em + 1;  int brd = A[sm];  int buf;  while (i < j)  {  while (A[--j] > brd);  while (A[++i] < brd);  if (i < j)  {  buf = A[j];  A[j] = A[i];  A[i] = buf;  };  }  return j;  }  int\* sortHoar(int A[], int sm, int em)  {  if (sm < em)  {  int hb = getHoarBorder(A, sm, em);  sortHoar(A, sm, hb);  sortHoar(A, hb + 1, em);  }  return A;  };  int main() {  setlocale(LC\_ALL, "Rus");  int size, i, j, sizeC = 0, ind = 0, ch = 3;  int A[100], B[100], C[100];  cout << "Количество элементов = ";  cin >> size;  for (i = 0; i < size; i++)  {  A[i] = rand() % 20;  if (A[i] % 2 != 0) {  C[ind] = A[i];  ind++;  sizeC++;  }  B[i] = rand() % 40;  if (B[i] % 2 == 0)  {  C[ind] = B[i];  ind++;  sizeC++;  }  }  cout << endl << endl;  cout << "Массив A: ";  for (i = 0; i < size; i++) cout << A[i] << " ";  cout << "\nМассив B: ";  for (i = 0; i < size; i++) cout << B[i] << " ";  cout << "\nМассив C: ";  for (i = 0; i < sizeC; i++) cout << C[i] << " ";  cout << "\nКак сортировать? 1 - Шелла, 2 - Хоар\n";  cin >> ch;  switch (ch) {  case 1: ShellSort(C, sizeC); break;  case 2: sortHoar(C, 0, sizeC - 1); break;  }  cout << "\nОтсортированный массив C: ";  for (i = 0; i < sizeC; i++) cout << C[i] << " ";  } |
| Результат в консоли |
|  |