Лаб 1

**Сортировка массивов**

Написать программу для сортировки массива из 50 элементов методом “пузырьковой” сортировки (Bubble Sort) или прямого выбора (Select Sort) (по вариантам). Массив считать из файла. Вывести на экран трудоемкость метода (количество сравнений).

**Вариант 3**

Метод “пузырьковой” сортировки.

Массив для сортировки:

292, 334, 172, 615, 45, 212, 136, 401, 274, 167, 566, 807, 495, 817, 457, 103, 309, 156, 480, 877, 528, 367, 967, 618, 547, 903, 258, 172, 33, 79, 601, 902, 199, 654, 11, 693, 233, 150, 774, 133, 661, 871, 500, 499, 707, 873, 921, 400, 935, 370

Лаб 2

**Графы. Поиск остова минимального веса.**

Написать программу, которая по алгоритму Краскала находит остов минимального веса для связного взвешенного неориентированного графа, имеющего 7 вершин. Граф задан матрицей весов дуг, соединяющих всевозможные пары вершин (0 означает, что соответствующей дуги нет). Данные считать из файла.

**Вариант 3**

$$\left(\begin{matrix}\begin{matrix}\begin{matrix}\begin{matrix}\begin{matrix}0\\0\\1\end{matrix}\\8\\0\end{matrix}\\12\\20\end{matrix}&\begin{matrix}\begin{matrix}\begin{matrix}\begin{matrix}0\\0\\15\end{matrix}\\2\\7\end{matrix}\\0\\21\end{matrix}&\begin{matrix}\begin{matrix}\begin{matrix}1\\15\\0\end{matrix}\\0\\3\end{matrix}\\11\\0\end{matrix}&\begin{matrix}\begin{matrix}\begin{matrix}8\\2\\0\end{matrix}\\0\\16\end{matrix}\\30\\5\end{matrix}\end{matrix}\end{matrix}&\begin{matrix}\begin{matrix}\begin{matrix}\begin{matrix}0\\7\\3\end{matrix}\\16\\0\end{matrix}\\4\\0\end{matrix}&\begin{matrix}\begin{matrix}\begin{matrix}12\\0\\11\end{matrix}\\30\\4\end{matrix}\\0\\10\end{matrix}&\begin{matrix}\begin{matrix}\begin{matrix}20\\21\\0\end{matrix}\\5\\0\end{matrix}\\10\\0\end{matrix}\end{matrix}\end{matrix}\right)$$

**Лаб 3**

**Графы. Нахождение кратчайшего расстояния между двумя вершинами с помощью алгоритма Форда-Беллмана**

Написать программу, которая по алгоритму Форда-Беллмана находит кратчайшее расстояние от указанной вершины до всех остальных вершин связного взвешенного неориентированного графа, имеющего 7 вершин (нумерация вершин начинается с 0). Граф задан матрицей весов дуг, соединяющих всевозможные пары вершин (0 означает, что соответствующей дуги нет). Данные считать из файла.

**Вариант 3**

**Вершина 2**

$$\left(\begin{matrix}\begin{matrix}\begin{matrix}\begin{matrix}\begin{matrix}0\\2\\3\end{matrix}\\7\\8\end{matrix}\\10\\1\end{matrix}&\begin{matrix}\begin{matrix}\begin{matrix}\begin{matrix}2\\0\\4\end{matrix}\\6\\12\end{matrix}\\19\\9\end{matrix}&\begin{matrix}\begin{matrix}\begin{matrix}3\\4\\0\end{matrix}\\16\\17\end{matrix}\\15\\11\end{matrix}&\begin{matrix}\begin{matrix}\begin{matrix}7\\6\\16\end{matrix}\\0\\18\end{matrix}\\21\\14\end{matrix}\end{matrix}\end{matrix}&\begin{matrix}\begin{matrix}\begin{matrix}\begin{matrix}8\\12\\17\end{matrix}\\18\\0\end{matrix}\\20\\12\end{matrix}&\begin{matrix}\begin{matrix}\begin{matrix}10\\19\\15\end{matrix}\\21\\20\end{matrix}\\0\\13\end{matrix}&\begin{matrix}\begin{matrix}\begin{matrix}1\\9\\11\end{matrix}\\14\\12\end{matrix}\\13\\0\end{matrix}\end{matrix}\end{matrix}\right)$$

**Лаб 4**

**Графы. Нахождение кратчайшего расстояния между двумя вершинами с помощью алгоритма Дейкстры**

Написать программу, которая по алгоритму Дейкстры находит кратчайшее расстояние от указанной вершины до всех остальных вершин связного взвешенного неориентированного графа, имеющего 6 вершин (нумерация вершин начинается с 0). Граф задан матрицей весов дуг, соединяющих всевозможные пары вершин (0 означает, что соответствующей дуги нет). Данные считать из файла.

**Вариант 3**

**Вершина 5**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **0** | **18** | **0** | **49** | **37** | **0** |
| **18** | **0** | **50** | **0** | **0** | **0** |
| **0** | **50** | **0** | **0** | **0** | **14** |
| **49** | **0** | **0** | **0** | **0** | **42** |
| **37** | **0** | **0** | **0** | **0** | **59** |
| **0** | **0** | **14** | **42** | **59** | **0** |

Лаб 5

**Задачи динамического программирования. Задача грабителя (задача “о рюкзаке”)**

Имеется склад, на котором присутствует некоторый ассортимент товаров. Запас каждого товара неограничен. У каждого товара своя стоимость Ci и масса mi. Написать программу, которая методом динамического программирования формирует такой набор товаров, чтобы его суммарная масса не превышала заданную грузоподъемность М, и стоимость была бы максимальной. На экран вывести промежуточные вычисления, сформированный набор, его стоимость и массу.

|  |  |  |  |
| --- | --- | --- | --- |
| Номер товара, i | mi | Ci | M |
| 1 | 11 | 9 | 47 |
| 2 | 9 | 8 |
| 3 | 12 | 3 |