

## Задания 1, 2. Внешняя сортировка (естественное и многопутевое слияние)

1. В текстовом файле задана целочисленная матрица размера  $n \times n$ . Отсортировать файл в порядке возрастания произведений элементов строк.
2. В текстовом файле задана целочисленная матрица размера  $n \times n$ . Отсортировать файл так, чтобы строки матрицы располагались **В** порядке убывания элементов, находящихся на главной диагонали.
3. В текстовом файле задана символьная матрица размера  $n \times n$ . Символы каждой строки матрицы образуют слово. Переформировать файл так, чтобы слова следовали в лексикографическом порядке.
4. В текстовом файле находятся  $m$  квадратных матриц. Создать выходной файл, в котором исходные матрицы расположены в порядке убывания произведений элементов главной диагонали каждой матрицы.
5. В текстовом файле заданы слова (не более 20), каждое из которых расположено на отдельной строке. Отсортировать файл так, чтобы слова следовали в обратном лексикографическом порядке (упорядочение от конца слова к началу).
6. Задан текстовый файл, каждая строка которого содержит слово и целое число. Отсортировать файл в порядке возрастания чисел.
7. В текстовом файле не более 30 строк. Отсортировать файл **В** порядке невозрастания количества различных букв в строках.
8. В текстовом файле находятся строки (не более 30). Отсортировать файл в порядке возрастания длин строк.
9. В текстовом файле задана прямоугольная матрица. Отсортировать файл в порядке возрастания произведений элементов строк матрицы.
10. Строка текстового файла имеет следующий вид: одна буква фамилии студента, один пробел, количество баллов по аттестации от 0 до 9. Отсортировать файл в порядке неубывания баллов.
11. В файле содержится информация о собаках различных пород: порода собаки, возраст, кличка, ФИО владельца, оценка, полученная собакой на последней выставке (удовл., хор., оч. хор., отл., нет). Переформировать файл следующим образом: упорядочить записи о собаках по возрастным группам (0-1.5 лет, 1.5 - 6, 6 - ...), а внутри каждой возрастной группы упорядочить записи по выставочным оценкам, полагая, что неучастие в выставке ("нет") приравнивается к самой низкой оценке.

## Задание 3. Алгоритмы с возвратом

### Граф

Задан граф, например, в виде пар вершин, которые соединены между собой.

1. Раскрасить граф минимальным количеством цветов, так чтобы дуга не соединяла две вершины одинакового цвета.

2. Найти все вершины графа, недостижимые из заданной вершины.
3. Определить является ли связанным заданный граф.
4. Найти вершину заданного графа, которая принадлежит каждому пути между двумя заданными вершинами.
5. Определить, изоморфны ли 2 графа.
6. Дан граф. Необходимо определить последовательность вершин, через которые необходимо пройти, чтобы получить этот граф, не открывая карандаша от бумаги и не проходя по одной и той же дуге несколько раз.
7. Из графа удалить вершины, недостижимые из заданной.

### **Дороги**

8. Задана система односторонних дорог. Найти путь, соединяющий города А и Б, не проходящий через заданное множество вершин.
9. Задана система двусторонних дорог. Найти замкнутый путь, проходящий через каждую вершину, и длиной не более 100 км.
10. Найти город в системе двусторонних дорог, у которого сумма расстояний до любого города минимальна.
11. По системе двусторонних дорог определить, определить есть ли в ней город, из которого можно добраться в любой другой менее чем за 100 км. Разрешается построить дополнительно 3 дороги.
12. По системе двусторонних дорог, определить можно ли закрыв какие-либо 3 из них, добиться того, чтобы из города А нельзя было попасть в город Б.
13. Задана система двусторонних дорог. N-периферией называется множество городов, расстояние от которых до выделенного города больше N. Определить Nпериферию для заданного N.

### **Геометрия**

14. Построить такой многоугольник (не обязательно выпуклый) с вершинами в заданном множестве, периметр которого максимален.
15. Найти минимальное множество прямых, на которых можно разместить все точки заданного множества.
16. В условии предыдущей задачи найти множество, содержащее ровно N точек, центр тяжести которого находится наиболее близко к началу координат.

## Задание 4. Trie-деревья

1. Процедура вставки задаваемого узла в trie — дерево и вывода дерева.
2. Удаление узла из trie — дерева и печать.
3. Подсчет количества слов, начинающихся с заданной приставки (с заданной последовательности букв).
4. Поиск слова в дереве и его печать .
5. Вставка в trie — дерево перевернутого слова и поиск слов по заданному окончанию.
6. В trie — дереве определить количество слов, содержащих букву «А».
7. В trie — дереве определить количество слов, которые содержат определенное количество согласных.
8. Из trie — дерева удалить все слова, которые содержат указанную подстроку.
9. В trie — дереве найти все слова, содержащие только буквы указанного множества.
10. Для заданного trie — дерева построить trie — дерево, в котором все слова расположены в обратном порядке.
11. Из trie — дерева удалить все слова с четной длиной.
12. Подсчитать в trie — дереве количество слов, оканчивающихся на согласную букву.

## Задание 5. Хеширование

1. Задан текст программы на любом языке. Определить частоту использования:
  - а. Идентификаторов
  - б. Ключевых слов
  - с. Символьных констант
2. Задан текстовый файл. Сформировать список слов, употребляющихся в тексте более пяти раз
3. Задан текстовый файл. Сформировать набор словосочетаний по два и более слова, встречающихся в тексте не менее двух раз.

4. Задан набор записей следующей структуры: табельный номер, ФИО, заработная плата. ПО табельному номеру найти остальные сведения
5. Задан набор записей следующей структуры: номер телефона, ФИО, адрес. По номеру телефона найти сведения о ФИО владельца и его адресе.
6. Задан набор записей следующей структуры: название кинофильма, режиссер, список актеров, краткое содержание. ПО заданному названию фильма найти остальные сведения.
7. Задан набор записей следующей структуры: номер автомобиля, его марка и ФИО владельца. По номеру автомобиля найти остальные сведения
8. Задан список имен людей. Определить частоту использования каждого имени в некотором тексте
9. Создать модуль, реализующий методы для работы с хеш-таблицей: инициализация, добавление элемента, удаление элемента, поиск. Кроме того, при заполнении хеш-таблицы выше заданного уровня размер хеш-таблицы должен автоматически увеличиваться
10. Исследовать зависимость числа коллизий от коэффициента заполненности хеш-таблицы
11. Из файла, содержащего текст на русском языке, слова помещаются в хеш-таблицы. Определить среднее число коллизий для нескольких заданных хеш-функций