

Лабораторная работа № 1 "Измерение времени выполнения алгоритма"

Цели: Выработать навыки оценки вычислительной сложности алгоритма.

Задачи: В лабораторной работе необходимо реализовать заданный алгоритм (см. таблицу ниже) и исследовать его вычислительную сложность. Алгоритм выбирается в соответствии с вариантом задания, полученным от преподавателя.

Вариант	Алгоритм/задача
1	Случайные блуждания без самопересечений
2	Поиск в массиве размера N смежного подмассива размера M с наибольшим средним значением элементов, посредством перебора всех подмассивов.
3	Сортировка выбором массива размера N
4	Исключение одинаковых элементов массива размера N
5	Определение количества "особых" элементов в матрице размера $N \times N$, таких что в строке слева от "особого" элемента находятся элементы меньшие его, а справа – большие.
6	В непустой последовательности слов, слова разделяются запятой, за последним словом стоит точка. Среди всех пар a_i и b_i , где a_i – первая, b_i – последняя буквы i – го слова последовательности, определить наиболее встречающуюся.
7	Сортировка перемешиванием массива размера N
8	Поиск в массиве размера N количества "троек" сумма которых равна 0.
9	Генерация всех возможных N -разрядных двоичных чисел
10	Сортировка расчёской массива размера N
11	Сортировка пузырьком массива размера N
12	Определение количества седловых точек в матрице размера $N \times N$. Элемент матрицы называется седловым, если он является наименьшим в своей строке и одновременно наибольшим в своем столбце.
13	Определить элементы в матрице $\ a_{ij}\ $ размера $N \times N$, как <i>true</i> , если значения i и j являются взаимно простыми и <i>false</i> – в противном случае.
14	Найти строку матрицы с максимальной суммой элементов
15	Поиск в целочисленном массиве размера N пары с наиболее близкими значениями.

Порядок работы:

1. Согласно варианту на ЯП (Java/Python/C#/...) в виде консольного приложения реализовать заданный алгоритм, убедиться в его корректности;
2. Доработать приложение так, чтобы на вход алгоритма подавались входные данные разного размера N , и для каждого размера измерялось время выполнения алгоритма. Полученная таблица с результатами (размер/время) должна сохраняться в файл в формате CSV.
3. открыть полученный CSV-файл в электронной таблице (MS Office Excel, OpenOffice Calc или подобной) и построить график зависимости времени выполнения алгоритма от размера входных данных;
4. выполнить теоретическую оценку эффективности алгоритма;
5. сделать выводы насчет соответствия полученных результатов теоретической оценке.

Методические указания

Изучение зависимости времени работы алгоритма от размера входных данных следует проводить следующим образом. Выбираются: начальное значение N_{start} , конечное значение N_{end} , приращение ΔN , количество повторений q . Далее перебираются все значения N , начиная с N_{start} до N_{end} с шагом ΔN , и для каждого N алгоритм выполняется q раз. Полученные q результатов сортируются по возрастанию, отбрасываются 20% максимальных значений, а для остальных 80% ищется среднее арифметическое. Полученное значение считается временем работы алгоритма на входных данных размера N .

Для измерения времени работы фрагмента кода следует использовать встроенные функции ЯП.

Результат измерений сохраняется в CSV-файле (см. Википедия энциклопедия. — <http://ru.wikipedia.org/wiki/CSV>). Этот текстовый формат предназначен для представления табличных данных. Каждая строка файла соответствует строке таблицы. Значения отдельных ячеек таблицы разделяются символом «;» (точка с запятой).

На рис. 1а приведен пример таблицы, а на рис. 1б – ее представление в формате CSV.

