

Задание 1. Эмпирический анализ временной сложности алгоритмов

Цель работы

Эмпирический анализ временной сложности алгоритмов

Задачи и методы

Для каждого n от 1 до 2000 произведите по пяти запускам замеры среднего машинного времени исполнения программ, реализующих нижеуказанные алгоритмы и функции. Изобразите на графике полученные данные, отражающие зависимость среднего времени исполнения от n . Проведите теоретический анализ временной сложности рассматриваемых алгоритмов и сравните эмпирическую и теоретическую временные сложности.

I. Сгенерируйте n -мерный случайный вектор $\mathbf{v} = [v_1, v_2, \dots, v_n]$ с неотрицательными элементами. Для \mathbf{v} реализуйте следующие вычисления и алгоритмы:

- 1) $f(\mathbf{v}) = \text{const}$ (постоянная функция);
- 2) $f(\mathbf{v}) = \sum_{k=1}^n v_k$ (сумма элементов);
- 3) $f(\mathbf{v}) = \prod_{k=1}^n v_k$ (произведение);
- 4) полагая, что элементы v – это коэффициенты многочлена P степени $n - 1$, вычислите значение $P(1,5)$ путем прямого (наивного) вычисления $P(x) = \sum_{k=1}^n v_k x^{k-1}$ (т.е. оценивая каждый член по одному) и методом Горнера представления многочлена в виде
$$P(x) = v_1 + x(v_2 + x(v_3 + \dots));$$
- 5) сортировка пузырьком (Bubble Sort) элементов \mathbf{v} ;
- 6) быстрая сортировка (Quick Sort) элементов \mathbf{v} ;
- 7) гибридная сортировка (Timsort) элементов \mathbf{v} .

II. Сгенерируйте случайные матрицы A и B размера $n \times n$ с неотрицательными элементами. Найдите обычное матричное произведение для A и B .

III. Опишите структуры данных и методы проектирования, которые использованы в рассматриваемых алгоритмах.