Домашнее задание #6, 02.11

1. (2б) Написать программу, которая для симметричной квадратной матрицы A вычисляет LL^T разложение или выводит $not\ positive\ definite$ если A не является положительно определенной.

2. (2б) Написать программу, которая по целому числу $1 \leq n \leq 10$, набору значений $x_1, \dots, x_m \in \mathbb{R}$, $n < m \leq 100$, и набору значений $y_1, \dots, y_m \in \mathbb{R}$ вычисляет многочлен P степени не больше n минимизирующий

$$\sum_{i=1}^{m} (P(x_i) - y_i)^2$$

   (1б*) Нарисовать график P и отметить на нем точки (x_i, y_i).

3. (2б) Написать программу, которая проверяет, что матрица $A \in \mathbb{R}^{n \times m}$ имеет линейно независимые строки (проанализировав $A^T A$) и минимизировать по x

$$||Ax - b||^2$$

4. Дан ориентированный сильно связный граф $G = \langle V, E \rangle$, $|V| = n$. Вася совершает случайное блуждание: изначально Вася равновероятно выбирает одну из вершин, после чего каждую секунду переходит в одну из соседних вершин. Находясь в вершине v Вася

   (1б*) С вероятнотью $1/d_v$ переходит в любую из вершин $u:\ (v, u) \in E$, $d_v = |\{u \mid (v, u) \in E\}|$.

   (2б*) С вероятнотью $\omega(v, u)/d_v$ переходит в вершину $u:\ (v, u) \in E$, $d_v = \sum_{(v,u) \in E} \omega(v, u)$, $\omega(v, u) > 0$.

   Для каждой вершины найдите вероятность нахождения в этой вершине после достаточно долгого блуждания.

   Предпочтительный формат.

   Входные данные. Граф в формате списка рёбер: строка с числами n m – количество вершин и ребер соответственно, затем m строк по два числа v_i u_i для первого пункта и по три числа v_i u_i $\omega(v_i, u_i)$ для второго, v_i, u_i – номера начальной и конечной вершины соответственно (индексация с 1).

   Выходные данные. n величин – вероятности нахождения в вершинах $1, 2, \dots, n$.

Замечание. Задачи с $*$ не обязательны к выполнению. Задачи $1-3$ будут зачтены только если решены с помощью LL^T разложения. Обратите внимание, что для положительно определенной матрицы A любой алгоритм нахождения LL^T корректно завершается. Задача 4 предполагает решение *однородной* системы, поэтому объявляется вне зачета. `numpy` содержит встроенную функцию для поиска разложения LL^T `numpy.linalg.cholesky`, эту и другие подобные функции можно использовать во всех заданиях кроме 1.