Куприянов Михаил, ПЗПИ-19-2

Ревью 1

B. Крысиные бега

Алгоритм решения

Используем стандартный алгоритм поиска количества путей фиксированной длины в графе. Возводим матрицу смежности графа в степень k с помощью бинарного возведения в степень, зачем считаем сумму элементов в первой строке.

Покажем по индукции что элемент a[i][j] для исходной матрицы возведенной в степень k равен количеству путей длины k из i d j. База: для k=1 a[i][j] равно количеству ребер из i в j, что и равно количеству путей длины 1. (для k=0 мы считаем ответом единичную матрицу: из вершины в саму себя один путь длины 0, из вершины в любую другую - ноль). Переход: пусть есть матрица b равная исходной матрице, возведенной в k-1 степень, для которой b[i][j] равно количеству путей длины k-1 из i в j, и исходная матрица c. Тогда, из определения произведения двух матриц: $a[i][j]=\sum_{l=1}^{n}(b[i][l]*c[l][j])$, покажем что в такой сумме мы считает каждый путь длины k, и при этом ровно один раз. Любой путь длины k из i в j может быть представлен как путь длины k-1 из i, и еще одно ребро, причем это последнее ребро заканчивается в j. Заметим что количества путей длины k-1 в вершину l по предположению индукции равно b[i][l], а количество последних ребер которые могут идти из l в j это просто количество ребер из l в j (c[l][j]). Тогда количество путей длины k из i в j с (k-1)-й вершиной l это b[i][l]*c[l][j], а сумма таких произведений для всех l и есть общее количество путей длины k.

Так как мы хотим посчитать количество путей длины k куда угодно из первой вершины то нам нужно посчитать сумму a[1][i] для каждой вершины i. Благодаря тому что перемножение матриц ассоциативно мы можем использовать бинарное возведение в степень.

Временная сложность

Мы считываем все ребра в матрицу за O(m), а затем возводим матрицу n*n в k степень. Одно перемножение матриц работает за n^3, всего, благодаря использованию бинарного возведения в степень log k перемножений. Итого **O(m+n^3*log k)** (затраты времени на суммирование всех чисел в одной строке: O(n) - незначительны).

Затраты памяти

На хранение одной матрицы нам необходимо n^2 памяти, всего мы создаем такие матрицы log k раз, на каждой итерации бинарного возведения в степень, итого **O(n^2*log k)** (однако можно организовать бинарное возведение в степень таким образом чтобы не создавать новые матрицы на каждой итерации и использовать всего O(n^2))