Теоретическое решение задачи С.

Немного перефразируем задачу. Комнаты будем рассматривать как вершины V графа $G = (V, E)$, а коридоры – его ребра E. Тогда становится понятно, что решение задачи сводится к поиску количества всевозможных путей фиксированной длины K из вершины 1 данного графа (по условию крысы бежали именно из вершины 1). Заметим, что в G допускаются кратные ребра, а также петли (следует из условия).

Хранить граф будем матрицей смежности $m[][]$ размера $N \times N$, в которой каждый элемент $m[i][j]$ будет отвечать за количество ребер из вершины i в вершину j (если таковых нет, то $m[i][j] = 0$).

Будем решать задачу методом динамического программирования. Пусть d_k будет матрицей ответов для путей длины k, $d_k[i][j]$ будет обозначать количество путей длины k от вершины i до j, а d_{k+1} – матрица ответов для путей на единицу больше.

База динамики: заметим, что в случае $k = 1$ ответом на задачу будет сумма всех значений первого ряда данной матрицы (т. к. ищем количество всех путей из вершины 1), а при $K = 0$ – сумма всех элементов первого ряда единичной матрицы, то есть ответ 1.

Переходы: Для каждой пары вершин i, j будем пробовать обновляться через другую вершину p. Тогда справедлива формула: $d_{k+1}[i][j] = \sum_{p=1}^{N} d_k[i][p] + d_k[p][j]$. Заметим, что данная формула, по сути, эквивалентна матричному произведению $d_k \times m$, соответственно можем сделать вывод, что $d_{k+1} = d_k \times m$. Таким образом задача сводится к перемножению матрицы смежности m саму на себя K раз $d_k = \prod_{i=1}^{K} m$. Также заметим, что в случае, если между вершинами u, v нет ребра, то очередное слагаемое не войдет в итоговую сумму, т. к. $d_k[u][v] = 0$, соответственно формула суммы корректна.

Теперь реализуем матричное возведение в степень. Чтобы делать это немного быстрее будем использовать бинарное возведение в степень (перемножение матриц ассоциативно, соответственно можем использовать данный алгоритм). Сами матрицы перемножать будем обычным стандартным алгоритмом произведения матриц.

После того, как динамика посчитана, ответом к задаче, как уже говорилось будет сумма элементов первого ряда результирующей матрицы $ans = \sum_{i=1}^{N} d_K[0]\,[i]$.

Оценим время работы описанного алгоритма.

1. Для считывания входных данных и заполнения матрицы смежности требуется время $O(N^2)$.
2. Алгоритм перемножения матрицы $m \times m$ для каждого ряда матрицы пробегает по очередному столбцу, суммирует значения и записывает в определенную ячейку. Время работы $O(N^3)$. Всего нужно перемножить матрицу саму на себя K раз, а бинарное возведение в степень в худшем случае работает за логарифм от размера степени K. Всего вызовов перемножений матриц будет $O(\log K)$, а соответственно суммарное время возведения матрицы m в K-степень будет равно $O(N^3 \cdot \log K)$.
3. Для того, чтобы посчитать ответ требуется пробежать по первому ряду итоговой матрицы и найти сумму. Время, требуемое на эту операцию – $O(N)$.

Таким образом, весь алгоритм отработает за $O(N^2 + N^3 \cdot \log K + N) = O(N^3 \cdot \log K)$ по времени.

Оценим требуемую память данного алгоритма. Требуется $O(N^2)$ для хранения исходной матрицы смежности m графа G и $O(N^2)$ для хранения матрицы результатов d_K. Соответственно всего требуется $O(N^2 + N^2) = O(N^2)$ памяти.