Shortcuts fixed length, number of tracks fixed length

The following describes these two objectives, built on the same idea: to reduce the problem to the construction of the power matrix (with the usual multiplication, and modified).

Number of paths of fixed length

Given an unweighted graph oriented $G$with $n$vertices, and given an integer k. Required for each pair of vertices $i$and $j$find the number of paths between these vertices consisting of exactly $k$the edges. Paths wherein arbitrary addresses are not necessarily simple (i.e., vertices may be repeated any number of times).

We assume that the graph is given **by the adjacency matrix** , ie matrix $g[][]$size $n \times n$, where each element is $g[i][j]$equal to one if between these vertices is an edge, and zero if there is no edge. Described below and the algorithm works in case of multiple edges if between some vertices $i$and $j$has immediately $m$edges, the adjacency matrix should record this number m. Algorithm also takes into account the correct loops in the graph, if any.

It is obvious that in this form **the adjacency matrix** of the graph is the **answer to the problem with**$k = 1$ - it contains a number of paths of length 1between each pair of vertices.

The decision will build **iteratively** let the answer for some $k$is found, we show how to build it $k + 1$. We denote $d_k$the matrix found answers for k, and through d_{k+1}- a matrix of responses you want to build. Then clear the following formula:

$$d_{k+1}[i][j] = \sum_{p=1}^{n} d_k[i][p] \cdot g[p][j].$$

Easy to see that recorded above formula - nothing more than the product of two matrices d_k, and $g$in the usual sense:

$$d_{k+1} = d_k \cdot g.$$

Thus, the **solution** of this problem can be represented as follows:

$$d_k = \underbrace{g \cdot \ldots \cdot g}_{k \text{ times}} = g^k.$$

It remains to note that the construction of the power matrix can be done efficiently using the algorithm **Up Binary exponentiation** .

Thus, the obtained solution has the asymptotic behavior $O(n^3 \log k)$lies in the construction of a binary k-th power of the adjacency matrix of the graph.

Shortcuts fixed length

Given a directed weighted graph $G$with $n$vertices, and given an integer k. Required for each pair of vertices $i$and $j$find the length of the shortest path between these vertices, consisting of exactly $k$the edges.

We assume that the graph is given **by the adjacency matrix** , ie matrix $g[][]$size $n \times n$, where each element $g[i][j]$contains the length of the edge from vertex $i$to vertex j. If the edges between any vertices not, then the corresponding element of the matrix to be equal to infinity ∞.

It is obvious that in this form **the adjacency matrix** of the graph is the **answer to the problem with**$k = 1$ - it contains the length of the shortest paths between each pair of vertices, or ∞if the length of the path 1does not exist.

The decision will build **iteratively** let the answer for some $k$is found, we show how to build it $k+1$. We denote $d_k$the matrix found answers for k, and through d_{k+1}- a matrix of responses you want to build. Then clear the following formula:

$$d_{k+1}[i][j] = \min_{p=1\ldots n} \left(d_k[i][p] + g[p][j] \right).$$

Look closely at this formula, it is easy to draw an analogy with the matrix multiplication: in fact, the matrix $d_k$is multiplied by the matrix g, only the multiplication operation instead of the sum of all $p$the minimum is taken over all p:

$$d_{k+1} = d_k \odot g,$$

where the operation $\odot$of multiplication of two matrices is defined as follows:

$$A \odot B = C \quad \Longleftrightarrow \quad C_{ij} = \min_{p=1\ldots n} \left(A_{ip} + B_{pj} \right).$$

Thus, **the solution** of this problem can be represented by this multiplication as follows:

$$d_k = \underbrace{g \odot \ldots \odot g}_{k \text{ times}} = g^{\odot k}.$$

It remains to note that the construction of the power of multiplication with this operation can be carried out efficiently by the algorithm **Up Binary exponentiation**as the only required for a property - associativity of multiplication - obviously there.

Thus, the obtained solution has the asymptotic behavior $O(n^3 \log k)$and is in the construction of binary k-th power of the adjacency matrix of the graph with the modified matrix multiplication.

Generalization to the case when the path length is required, not more than a predetermined length

Above solutions solve problems when you need to consider ways to certain fixed length. However, these solutions can be adapted to solve problems and when required to consider paths that contain **no more** than a specified number of edges.

You can do this by modifying the input bit count. For example, if we are only interested in the path terminating at a particular vertex t, then the graph can **add loop** (t, t) zero weight.

If we are still interested in answers for all pairs of vertices, then simply adding loops to all vertices spoil answer. Instead, you can **be bisected** each vertex: for each vertex $v$to create an additional vertex v', edge hold $(v, v')$and add a loop (v', v').

Deciding on a modified graph problem of finding ways to fixed-length answers to the original problem will be obtained as answers between the vertices $i$and j'(ie additional vertices - the vertices-end, in which we can "whirl" as many times).