Floyd算法思想

1)算法思想原理：

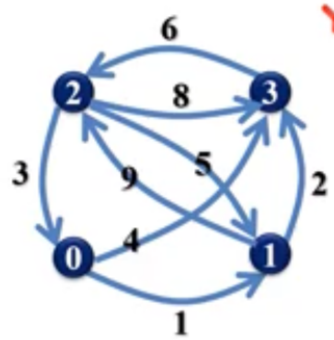
Floyd算法是一个经典的动态规划算法。用通俗的语言来描述的话，首先我们的目标是寻找从点i到点j的最短路径。从动态规划的角度看问题，我们需要为这个目标重新做一个诠释（这个诠释正是动态规划最富创造力的精华所在）

从任意节点i到任意节点j的最短路径不外乎2种可能，1是直接从i到j，2是从i经过若干个节点k到j。所以，我们假设Dis(i,j)为节点u到节点v的最短路径的距离，对于每一个节点k，我们检查Dis(i,k) + Dis(k,j) < Dis(i,j)是否成立，如果成立，证明从i到k再到j的路径比i直接到j的路径短，我们便设置Dis(i,j) = Dis(i,k) + Dis(k,j)，这样一来，当我们遍历完所有节点k，Dis(i,j)中记录的便是i到j的最短路径的距离。

Floyd算法示例演示

$$A^{(0)} = \begin{vmatrix} 0 & 1 & \infty & 4 \\ \infty & 0 & 9 & 2 \\ 3 & 4 & 0 & 7 \\ \infty & \infty & 6 & 0 \end{vmatrix}$$

$$path^{(0)} = \begin{vmatrix} 00 & 01 & 02 & 03 \\ 10 & 11 & 12 & 13 \\ 20 & 201 & 22 & 203 \\ 30 & 31 & 32 & 33 \end{vmatrix}$$



Floyd算法示例演示

$$A^{(1)} = \begin{vmatrix} 0 & 1 & 10 & 3 \\ \infty & 0 & 9 & 2 \\ 3 & 4 & 0 & 6 \\ \infty & \infty & 6 & 0 \end{vmatrix}$$

$$path^{(1)} = \begin{vmatrix} 00 & 01 & 012 & 013 \\ 10 & 11 & 12 & 13 \\ 20 & 201 & 22 & 2013 \\ 30 & 31 & 32 & 33 \end{vmatrix}$$

Floyd算法示例演示

$$A^{(2)} = \begin{vmatrix} 0 & 1 & 10 & 3 \\ 12 & 0 & 9 & 2 \\ 3 & 4 & 0 & 6 \\ 9 & 10 & 6 & 0 \end{vmatrix}$$

$$path^{(2)} = \begin{vmatrix} 00 & 01 & 012 & 013 \\ 120 & 11 & 12 & 13 \\ 20 & 201 & 22 & 2013 \\ 320 & 3201 & 32 & 33 \end{vmatrix}$$

Floyd算法示例演示

$$A^{(3)} = \begin{vmatrix} 0 & 1 & 9 & 3 \\ 11 & 0 & 8 & 2 \\ 3 & 4 & 0 & 6 \\ 9 & 10 & 6 & 0 \end{vmatrix}$$

$$path^{(3)} = \begin{vmatrix} 00 & 01 & 0132 & 013 \\ 1320 & 11 & 132 & 13 \\ 20 & 201 & 22 & 2013 \\ 320 & 3201 & 32 & 33 \end{vmatrix}$$