

Floyd-Warshall 算法

【算法概述】

假如我们要找任意有向图或无向图中两个点之间的最短距离，使用Dijkstra算法或者Bellman Fold算法时间复杂度为 $O(n^3)$ ，时间包括求源点到任意目标点之间的最短距离 $O(n^2)$ 和遍历图所有点以求出任意两个点的距离 $O(n)$ 。

【算法思路】

Floyd-Warshall采用动态规划的思想。

算法首先标记所有顶点为1，。。n。假设 $W[i, j]^k$ 为顶点i和j之间最短距离，而k代表i与j之间顶点数。

初始化时我么令 $k = 0$ ， 据此我们根据图构造一个连接矩阵，矩阵的值代表两个顶点之间的距离。

状态转移方程为：

$$W[i, j] = \min(W[i, j]^{k-1}, W[i, k]^{k-1} + W[k, j]^{k-1})$$

【伪代码】

```
1 let dist be a  $|V| \times |V|$  array of minimum distances initialized to  $\infty$  (infinity)
2 for each vertex  $v$ 
3    $\text{dist}[v][v] \leftarrow 0$ 
4 for each edge  $(u, v)$ 
5    $\text{dist}[u][v] \leftarrow w(u, v)$  // the weight of the edge  $(u, v)$ 
6 for  $k$  from 1 to  $|V|$ 
7   for  $i$  from 1 to  $|V|$ 
8     for  $j$  from 1 to  $|V|$ 
9       if  $\text{dist}[i][j] > \text{dist}[i][k] + \text{dist}[k][j]$ 
10          $\text{dist}[i][j] \leftarrow \text{dist}[i][k] + \text{dist}[k][j]$ 
11       end if
```