

# Dijkstra 算法 (最短路径)

1. 算法 (1) ~~在~~ 还未求出到选取点最短<sup>距离</sup>的点的集合 (记为  $V-U$ ) 中选取距离最小的点加入集合  $U$ , 点记为  $V_{min}$ .

(2) 修正。如果  $V_{min}.length + arcs[min][i] < V_i.length$ , 则把  $V_i.length$  改为较小的, 且  $V_i$  的前趋顶点改为  $V_{min}$ 。(其中  $V_i$  是  $V-U$  中剩下的点).

## 2. 内核:

```

void dijkstra (graph, dist[])
{
    // (默认为求到第一个顶点的
    // 路径) (dist[]).
    // (dist[] 初始化)
    for (i=1; i < graph.n; i++) {
        minw = MAX; min = 0;
        // ① 选
        for (j=1; j < graph.n; j++)
            if ((graph.arcs[j][i] == 0) && (dist[j].length < minw)) {
                minw = dist[j].length; min = j;
            }
        if (min == 0) break;
        graph.arcs[min][min] = 0; // (加入 U)
        // ② 修正
        for (j=1; j < graph.n; j++)
            if ((graph.arcs[j][min] != 0) && (dist[j].length > dist[min].length + graph.arcs[min][j])) {
                dist[j].length = dist[min].length + graph.arcs[min][j];
                dist[j].prev = min;
            }
    }
}

```

## 3. 图示

