导航

- 首页
- 社区主页
- 当前事件
- 最近更改
- 随机页面
- 使用帮助
- NOCOW地图
- 新手试练场

搜索

进入 搜索

工具箱

- 链入页面
- 链出更改
- 特殊页面
- 可打印版
- 永久链接

Johnson算法

算法简介 [编辑]

Johnson算法适用于求All Pairs Shortest Path. Johnson算法应用了重标号技术，先进行一次Bellman-Ford算法，然后对原图进行重标号，w'(i,j)=h[i]-h[j]+w(i,j)。 然后对每个点进行一次Dijkstra，每次Dijkstra的复杂度为O(nlogn+m)，于是算法复杂度为O(n^2logn+m)。

扩充 [编辑]

Johson算法是目前最高效的在无负环可带负权重的网络中求所有点对最短路径的算法. Johson算法是Bellman-Ford算法, Reweighting(重赋权重)和Dijkstra算法的大综合. 对每个顶点运用Dijkstra算法的时间开销决定了Johnson算法的时间开销. 每次Dijkstra算法(d堆PFS实现)的时间开销是O(E * lgd(V)). 其中E为边数, V为顶点数, d为采用d路堆实现优先队列ADT. 所以, 此种情况下Johnson算法的时间复杂度是O(V * E * lgd(V)).

图论及图论算法 [编辑]

图 - 有向图 - 无向图 - 连通图 - 强连通图 - 完全图 - 稀疏图 - 零图 - 树 - 网络

基本遍历算法：宽度优先搜索 - 深度优先搜索 - A* - 并查集求连通分支 - Flood Fill

最短路：Dijkstra - Bellman-Ford（SPFA） - Floyd-Warshall - Johnson算法

最小生成树：Prim - Kruskal

强连通分支：Kosaraju - Gabow - Tarjan

网络流：增广路法（Ford-Fulkerson，Edmonds-Karp，Dinic） - 预流推进 - Relabel-to-front

图匹配 - 二分图匹配：匈牙利算法 - Kuhn-Munkres - Edmonds' Blossom-Contraction

Johnson算法是一个小作品，欢迎帮助扩充这个条目。

2个分类：图论 | 小作品