

Contents

- [Dijkstra算法参考](#)
- [辅助函数--队列](#)
- [准备数据](#)
- [主脚本：测试图算法](#)

Dijkstra算法参考

第一种方法，比较倾向于计算机算法

大卫.M.史密斯著，卓金武译，《MATLAB工程计算》第三版，机械工业出版社2018年

主要函数

```
% function D = grDijkstra(A,home,target)
% % 输入为邻接矩阵，起始节点的索引，终止节点的索引
% % 此处显示详细说明
% pq= pqEnq([],Path(home,0)); % 初始化优先队列，包含起始节点，成本为0
% while ~isempty(pq)
%     [pq current]=qDeq(pq); %此时，队列包含一个路径结构体
%     if pthGetLast(current) == target
%         D=sparse(0);
%         answer =current.nodes;
%         for ans=1:length(answer)-1
%             D(answer(ans),answer(ans+1)) =1;
%         end %如果出队的节点是target，则该函数将创建一个新的邻近矩阵，表示从home节点
%             % 到target节点的路径
%         return;
%     end % if last(current)==target
%     endnode = pthGetLast(current);
%     children=A(endnode,:); %%检索其子节点
%     children =find(children ~=0);
%     for achild =children %遍历这些子节点，检查它们是否在当前路径上，当一个子节点
%         % 入队时，将它添加到当前路径的末尾，同时，整个路径被插入队列
%         len =A(endnode,achild);
%         if ~any(achild == current.nodes)
%             gclone = Path( [current.nodes achild] , current.key+ len );
%             pq=pqEnq(pq,gclone);
%         end
%     end
% end
% D=[];
% end
%
```

辅助函数--队列

```
% function pq =pqEnq(pq,item)
% %按顺序入队
% % 优先队列的enqueue函数
% in=1;
% at= length(pq)+1;
% while in<= length(pq)
%     if is_before(item,pq{in})
%         at =in;
%         break;
%     end
%     in=in+1;
% end
% pq =[pq(1:at-1) {item} pq(at:end)];
```

```

% end
%
%
%
%
% function ans =is_before(a,b)
% % 比较两个对象，
% acl=class(a);
% ans=false;
% if isa(b,acl)
%     switch acl
%         case 'double'
%             ans= a<b;
%         case 'struct'
%             if isfield(a,'key')
%
%                 ans=a.key<b.key;
%             elseif isfield(a,'dod')
%                 ans=age(a)<age(b);
%             else
%                 error(' 不同的结构没法比较! ')
%             end
%         otherwise
%             error(['can ' 't compare ' acl 's'])
%     end
% end
% end

% function [q ans]=qDeq(q)
% % 出队
% ans=q{1};
% q=q(2:end);
% end

```

```

% function A= graAdjacency(node,cost,dir)
% % 将图形数据node,cost,direction数组转换为邻接矩阵
% [m,cols]=size(node)
% n=length(cost);
% k=0; %稀疏矩阵元素个数
% for is= 1:n %遍历边的列表
%     iv=0; % 初始化连接到该边的节点数
%     for ir=1:m %遍历节点数组的节点
%         for ic=1:cols % 遍历节点数组的列，查找连接到该边的节点
%             if node(ir,ic) == is
%                 iv= iv+1;
%                 if iv>2 %每条边只有两个端点，如果大于2，表示是一个错误的数据集
%                     error(' 错误的交叉矩阵');
%                 end
%                 ij(iv) =ir; % 将各个端点保存到局部变量ij中
%             end
%         end
%     end
%     if iv~=2 % 当遍历完成时，边的两端必须都有一个节点
%         error(sprintf(' 找不到边 %d 的终端' ,is));
%     end
%     t=cost(is); %获取边的权重
%
%     if dir(is) ~= -1 % 双向边必须在矩阵中出现两次，检查该边是双向还是正向的
%         k=k+1;
%         ip(k)=ij(1);
%         jp(k)=ij(2);
%
%         tp(k)=t;
%     end
% end

```

```

%      end
%      if dir(is) ~= 1 % 同样，如果边不是正向的，输入反向路径
%          k=k+1;
%          ip(k)=ij(2);
%          jp(k)=ij(1);
%          tp(k)=t;
%      end
% end
% A=sparse(ip, jp, tp); % 构造稀疏矩阵
% end

```

准备数据

% makeGraph.m

```

cost=[2 2 2 2 2 3 3 3 3 1 2 1 3]; %边的权重
dir=[2 2 2 2 2 2 2 2 2 2 2 2 2]; %边的方向
node=[1 2 3 4 5;1 6 7 0 0;2 7 8 0 0 ;3 8 9 0 0;4 11 13 9 0;5 6 10 0 0;10 11 12 0 0;12 13 0 0 0];
% % 边的序列为: A-B, A-C, A-D, A-E, A-F, B-F, B-C, C-D, D-E, F-G, E-G, G-H, E-H
% %连接node的每一行元素, 指分别从A, B, ..., H开始的边
coord=[5 6 ;3 9;1 6 ;3 1;6 2;6 8;9 7;10 2]; %A, B, ..., H的坐标, 只在画图时用
A=graAdjacency(node, cost, dir) %函数调用
%

```

m =

8

cols =

5

A =

(2, 1)	2
(3, 1)	2
(4, 1)	2
(5, 1)	2
(6, 1)	2
(1, 2)	2
(3, 2)	3
(6, 2)	3
(1, 3)	2
(2, 3)	3
(4, 3)	3
(1, 4)	2
(3, 4)	3
(5, 4)	3
(1, 5)	2
(4, 5)	3
(7, 5)	2
(8, 5)	3
(1, 6)	2
(2, 6)	3
(7, 6)	1
(5, 7)	2
(6, 7)	1
(8, 7)	1
(5, 8)	3

主脚本：测试图算法

```
%dijkstra_main.m 此脚本由于有input交互式函数，需要单独放一个程序文件中单独运行

% start =1;
% while start >0
%     gplot(A, coord, 'ro-')
%     hold on
%     for index =1:length(coord)
%         str =char('A' +index -1);
%         text(coord(index,1)+0.2, coord(index,2)+0.3, str);
%     end
%     axis([0,11,0,10]);axis off ;hold on
%     ch =input(' 起始节点(大写字母A):','s');
%
%     start =ch- 'A' +1;
%     if start>0
%         ch=input(' 目标节点(H): ','s')
%         target =ch- 'A' +1;
%         disp(' 原图');pause
%
%
%     D=grDijkstra(A, start, target);
%
%     gplot(D, coord, 'bo-')
% end
% end
```