计算机问题求解 – 论题3-7
- 树

2016年10月12日

Theorem 4.1 *An edge e of a graph G is a bridge if and only if e lies on no cycle of G*.

推论：Every edge in a tree is a bridge

问题1：

从应用的角度看，上述推论有何指导意义？

For an n-vertex graph G (with $n \geq 1$), the following are equivalent (and characterize the trees with n vertices).

A) G is connected and has no cycles.

B) G is connected and has $n-1$ edges.

C) G has $n-1$ edges and no cycles.

D) For $u, v \in V(G)$, G has exactly one u, v-path.

问题2:

如何证明一系列命题等价?

问题3:

不能有回路对最小顶点次数有什么影响?

树的性质的“极限性”

a) Every edge of a tree is a cut-edge.

b) Adding one edge to a tree forms exactly one cycle.

c) Every connected graph contains a spanning tree.

问题4：

什么样的图生成树是唯一的，为什么？

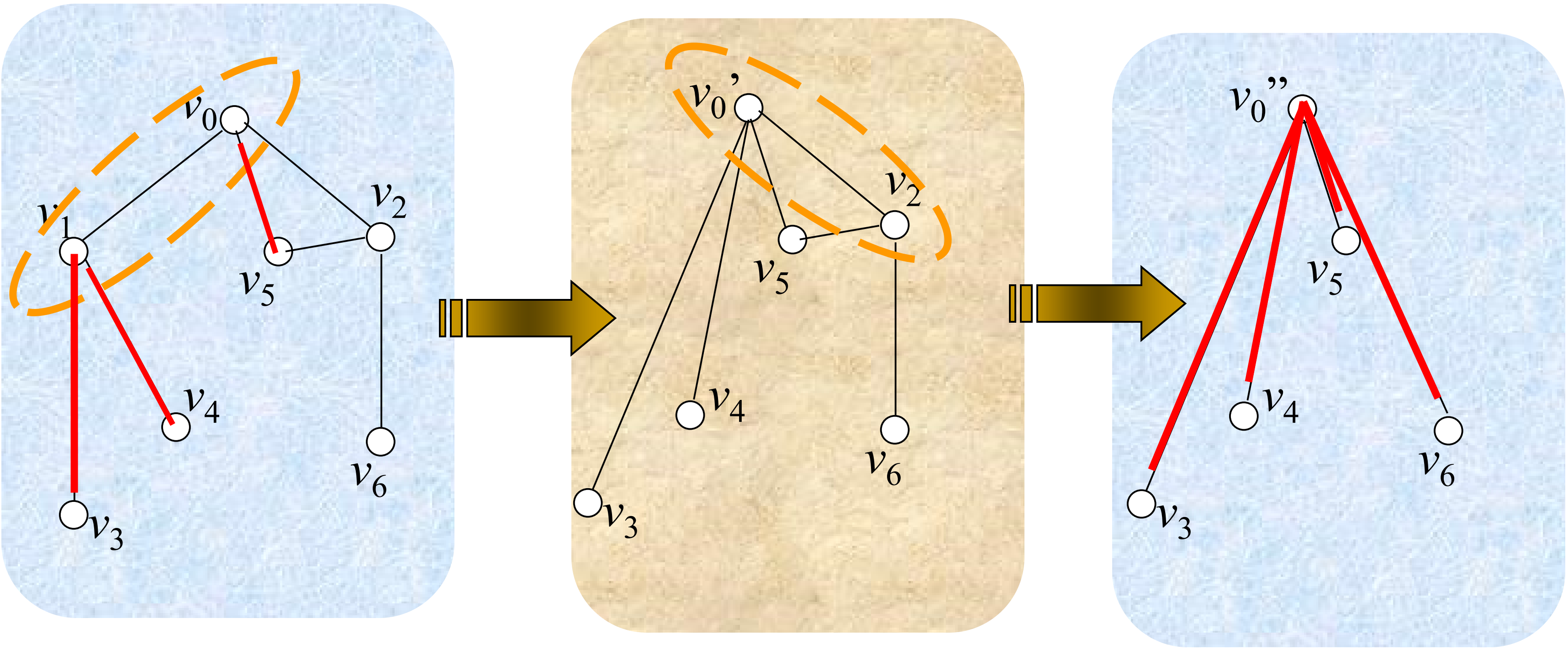
Theorem 4.4 *Every tree of order n has size n* - 1.

问题：有n-1条边的n点连通图，一定是树？
有n-1条边的无环连通图，一定连通n个点？

问题5：如何构造一个连通图的生成树？

无向连通图遍历算法一定得到一棵树

Merging Two Vertices



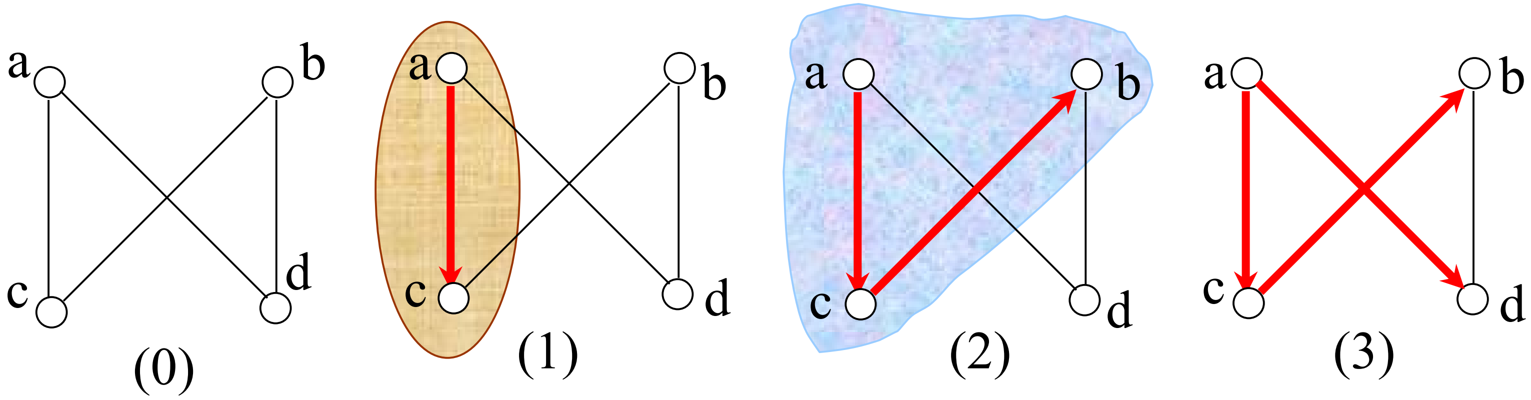
Matrix Operation for Merging

$$
\begin{array}{cc}
 & \begin{matrix} v_0 & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{matrix} \\
\begin{matrix} v_0 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{matrix} &
\begin{bmatrix}
0 & 1 & 1 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0
\end{bmatrix}
\end{array}
\qquad
\begin{array}{cc}
 & \begin{matrix} v_0' & v_2 & v_3 & v_4 & v_5 & v_6 \end{matrix} \\
\begin{matrix} v_0' \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{matrix} &
\begin{bmatrix}
0 & 1 & 1 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0
\end{bmatrix}
\end{array}
\qquad
\begin{array}{cc}
 & \begin{matrix} v_0'' & v_3 & v_4 & v_5 & v_6 \end{matrix} \\
\begin{matrix} v_0'' \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{matrix} &
\begin{bmatrix}
0 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0
\end{bmatrix}
\end{array}
$$

Merging v_0 and v_1

Merging v_0' and v_2

Constructing a Spanning Tree



0. Let a be the starting vertex, selecting edges one by one in original *R*

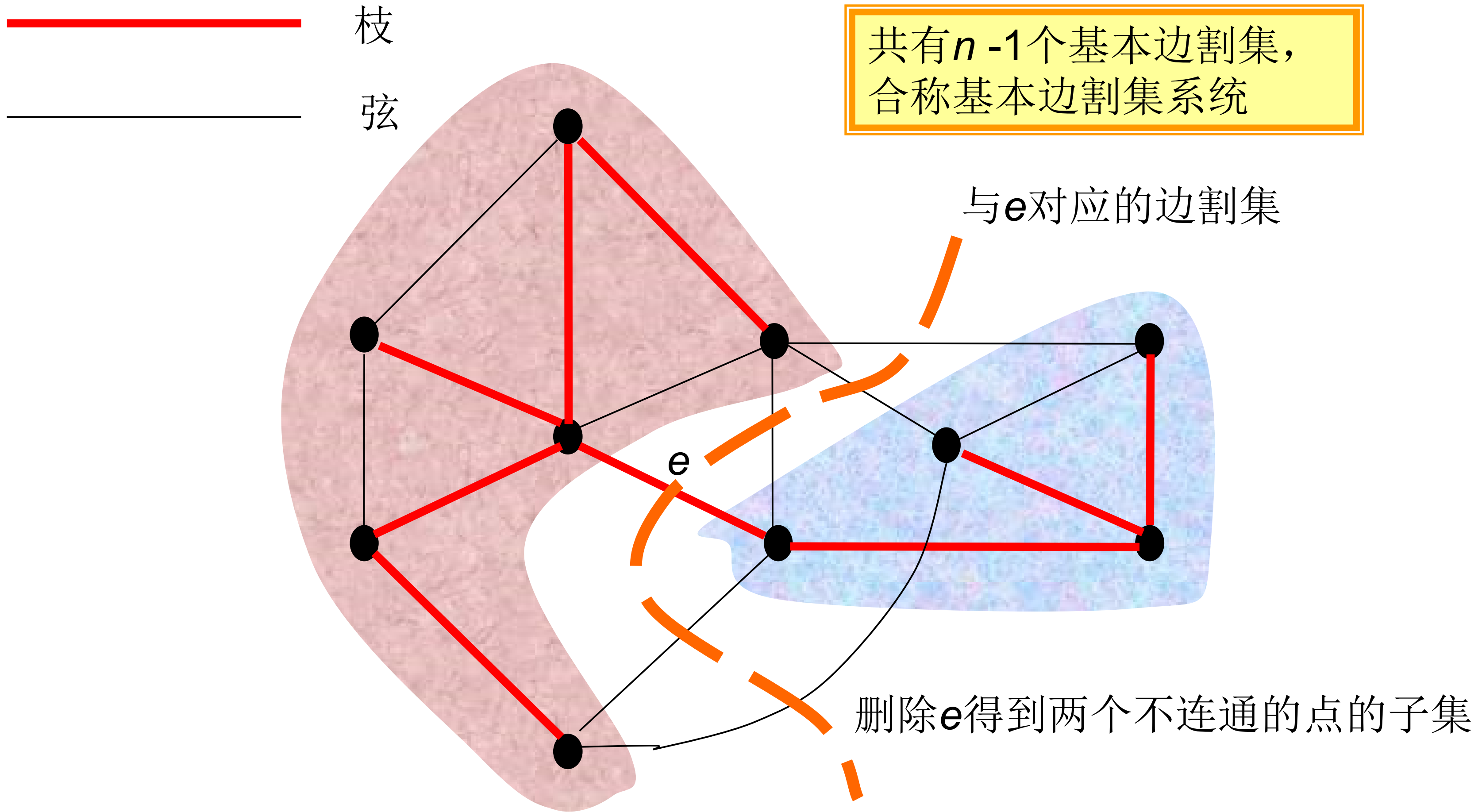
1. Merging a and c into a'({a,c}), selecting (a,c)

2. Merging a' and b into a"({a,c,b}), selecting (c,b)

3. Merging a" and d into a"'({a,c,b,d}), selecting (a,d) or (d,b)

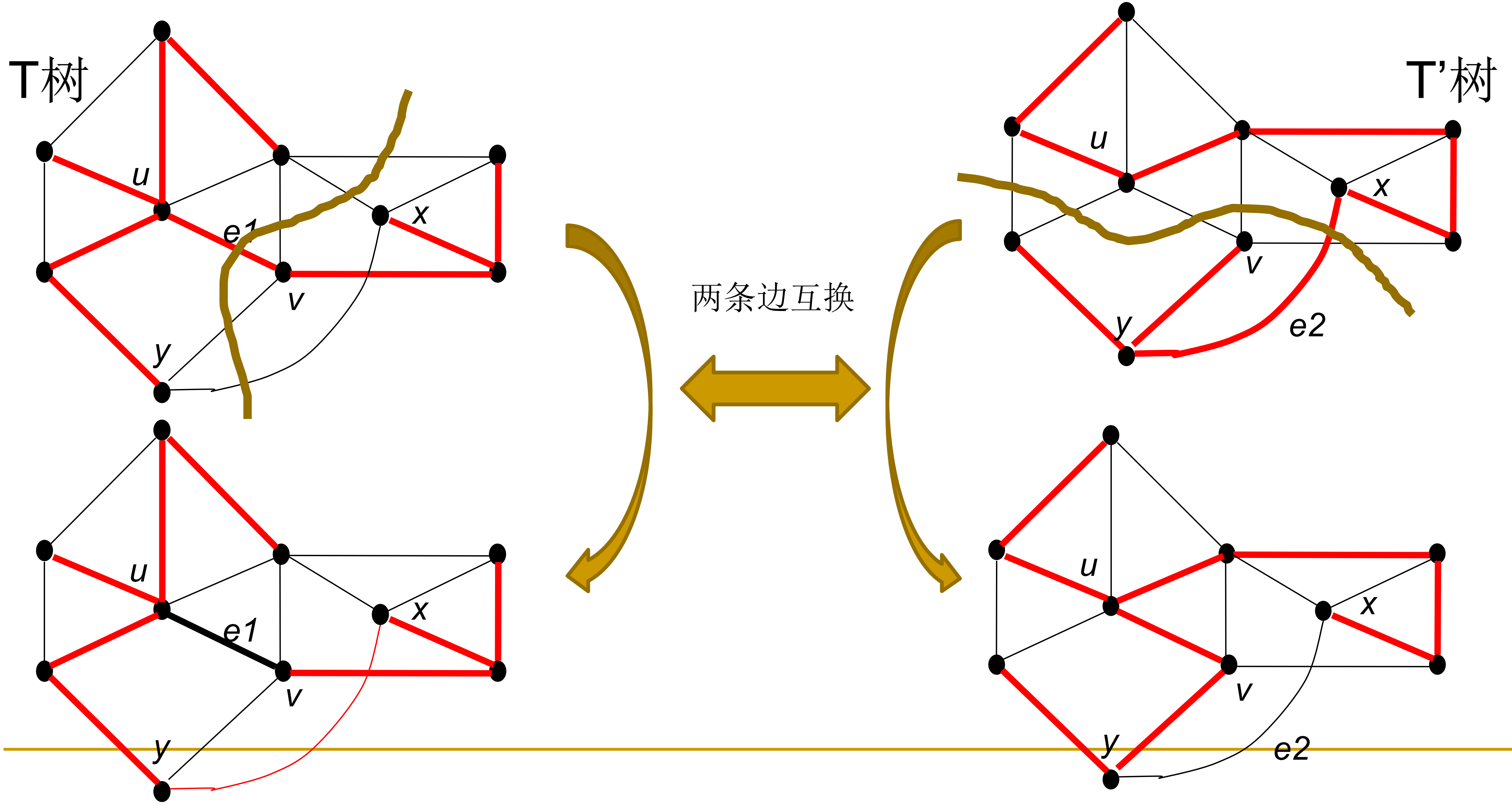
Ending, as only one vertex left

生成树：树“里”与树“外”的边



问题6：如何寻找一个连通图最“薄弱”的地方？

暗藏玄机：两棵不同的生成树



生成树的“变换”

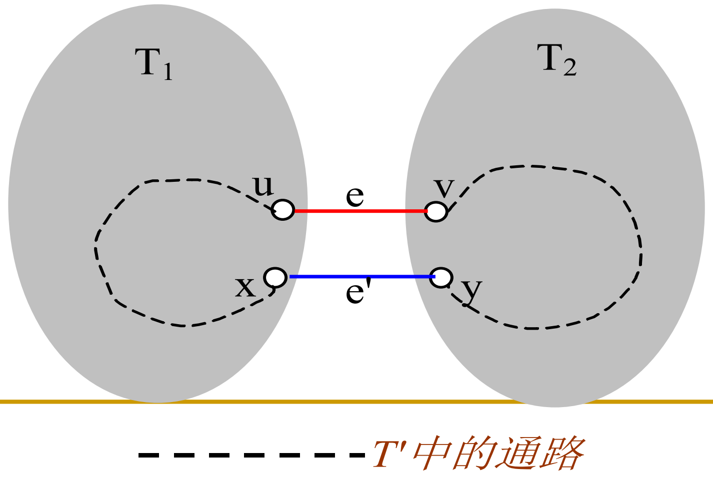
- 定理：T与T'均是图G的生成树，存在$e \in E_T$，但$e \notin E_{T'}$，证明：必有$e' \in E_{T'}$，但$e' \notin E_T$，满足：T-{e}∪{e'}和T'-{e'}∪{e}均是G的生成树。

  - 证明概要：

    设e=uv, T-{e}必含两个连通分支，设为T_1, T_2。

    ∵T‘是连通图，T’中有uv-通路，其中必有一边满足其两个端点x,y分别在T_1，T_2中，设其为e‘，显然T-{e}∪{e’}是生成树。

    而T‘∪{e}必含唯一回路，且该回路中必定包含e’。将e’从该回路中删去，得到T*=T‘-{e’}∪{e}。显然，T'-{e'}∪{e}是生成树。



问题7：这个定理会被用于证明最小生成树算法的正确性，你能大致推测出怎么用吗？

问题8：

什么是最小生成树问题？
它有什么实际应用？

Generic Algorithm for MST Problem

Input: G: a connected, undirected graph

w: a function from V_G to the set of real number

Generic-MST(G,w)

$A \leftarrow 0$

while A does not form a spanning tree

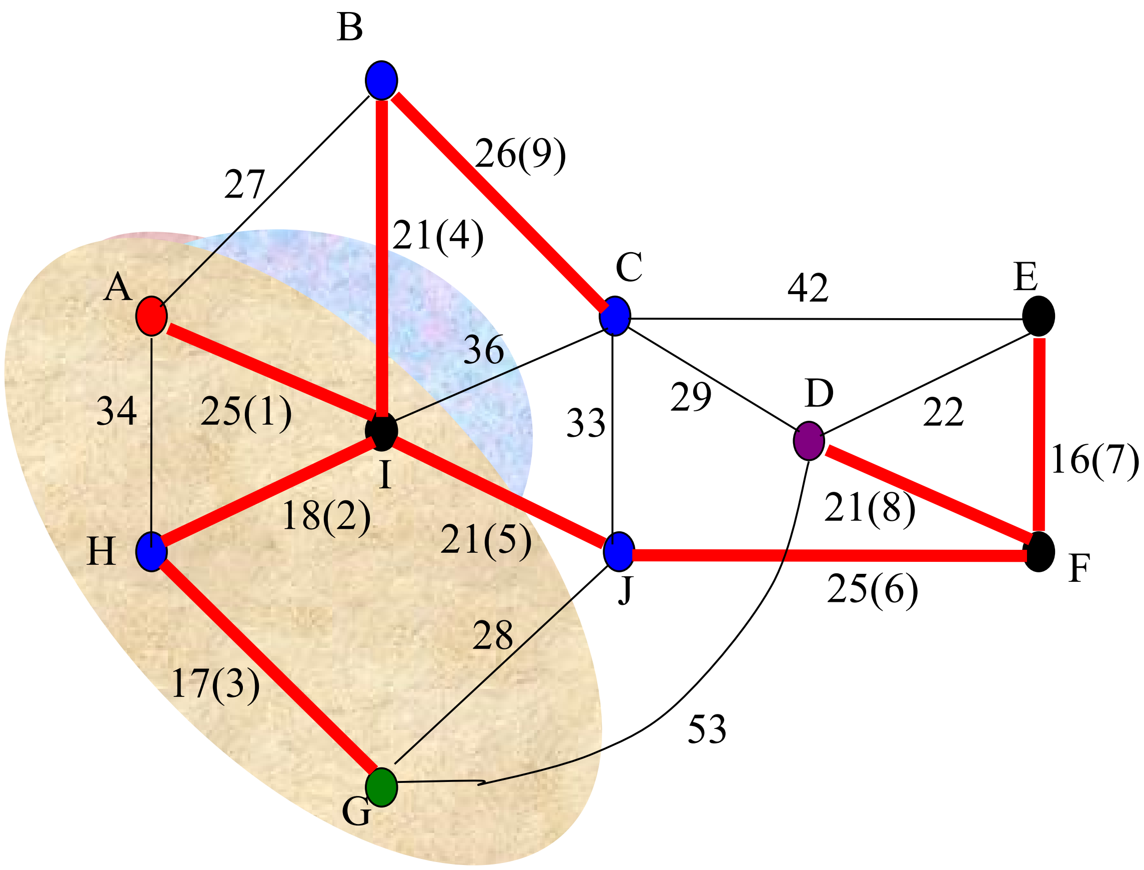
    do find an edge (u,v) that is safe for A

    $A \leftarrow A \cup \{(u,v)\}$

return A

Output: a minimal spanning tree of G

Prim's Algorithm for MST



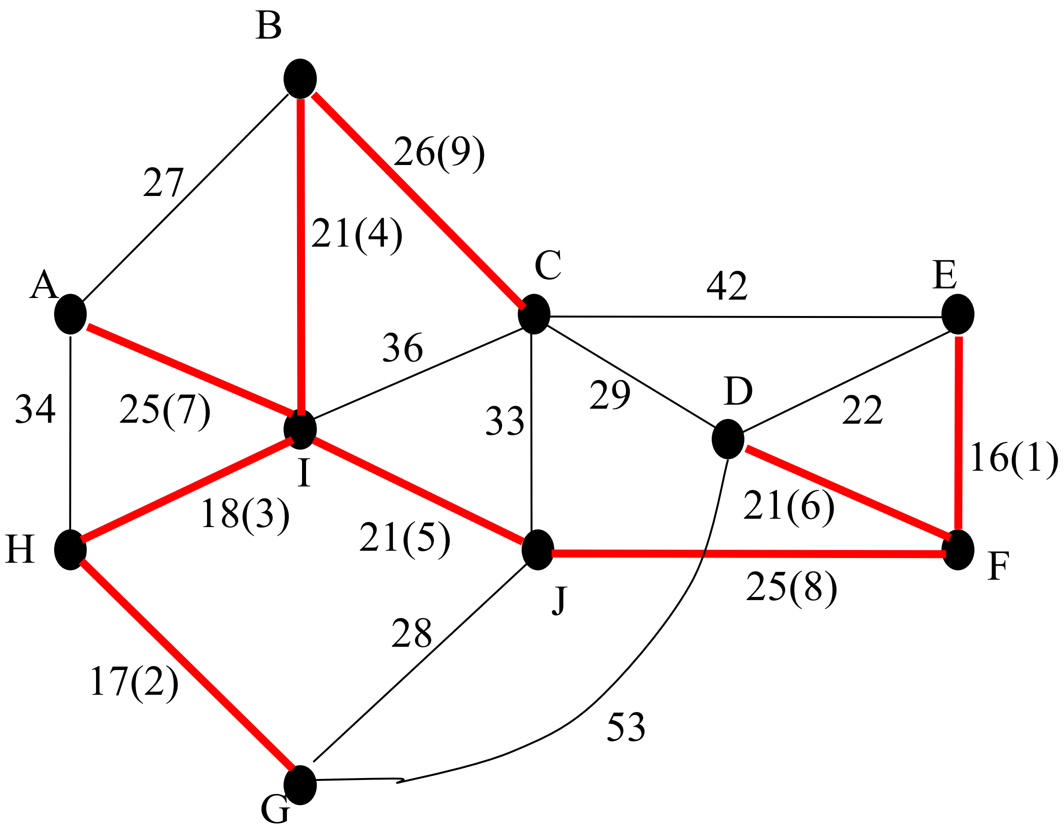
Step 1: V={A}, E={}

Step 2: Select the nearest neighbor of V, u, add the edge connecting u and some vertex in V into E

Step 3: Repeat step 2 until E contains n-1 edges

End of Algorithm

Kruskal's Algorithm for MST



Step 1: $E=\{\}$

Step 2: Select the edge with the least weight, and not making a cycle with members of E

Step 3: Repeat step 2 until E contains n-1 edges

End of Algorithm

How to prove your greedy choice property?

Of course, we must prove that a greedy choice at each step yields a globally optimal solution. Typically, as in the case of Theorem 16.1, the proof examines a globally optimal solution to some subproblem. It then shows how to modify the solution to substitute the greedy choice for some other choice, resulting in one similar, but smaller, subproblem.

Theorem 4.12 *Kruskal Algorithm produces a minimum spanning tree in a connected weighted graph.*

证明要点：

1,假设算法得到的T不是最小生成树，找一个最小生成树T'，尝试发现矛盾；

2,这个T'有其特殊性：将T的边按权不减序(也是算法选择的序)排好，将所有最小生成树也按边权不减序排列，取和T种e1,e2,…,ek相同的树中k最大的最小生成树T';

3,e_{k+1}是在T中但不在T'中的最小边；

4,针对e_{k+1}，必定存在T'中的边e',交换e_{k+1}和e'，得到新树T''。按照算法，$w(e') >= w(e_{k+1})$.

5, $w(T'') = w(T') - w(e') + w(e_{k+1})$. T''也是最小生成树。但同时，因T'最小，所以，$w(e') = w(e_{k+1})$.

6,T''的前k+1条边和T相同。矛盾。

问题9：

你认为要实现上述两个算法，分别需要怎样的数据结构？

问题10：

关于上述两个算法的时间复杂度你能说些什么吗？

问题11：

两个算法解决同样的问题，为什么两个都很有用？关于树的遍历也有类似情况，有什么不同吗？

Open Topics:

1,如果用相邻矩阵来表示一个图，你如何判断这个图是否是树？

2,设计一种用相邻矩阵表示权图的方案，并在这个方案基础上设计一个构造最小生成树的算法。