

生成树

回顾

2

- 内容1：表达式的（逆）波兰记法
- 内容2：二叉搜索树
- 内容3：前缀码与Huffman编码

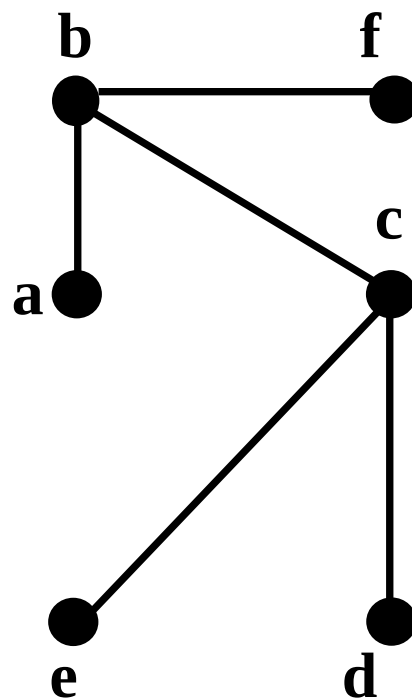
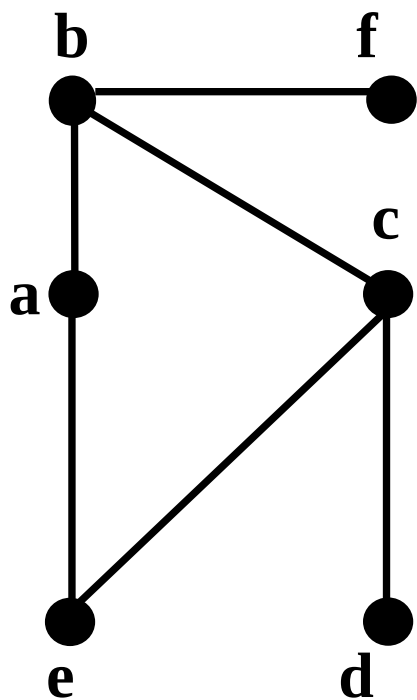
本节提要

- 内容1：生成树及其构造方法
- 内容2：最小生成树算法

生成树

- 定义：若图 G 的生成子图是树，则该子图称为 G 的生成树。
 - ▣ 注：含有 G 的所有顶点的子图称为 G 的生成子图
- 无向图 G 连通 当且仅当 G 有生成树
 - ▣ 证明(充分性显然):
 - ⇒ 注意：若 G 是有简单回路的连通图，删除回路上的一条边， G 中的回路一定减少。(因此，用“破圈法”总可以构造连通图的生成树)
- 简单无向图 G 是树 当且仅当 G 有唯一的生成树。

构造生成树：深度优先搜索



深度优先搜索算法

Procedure DFS(G : 带顶点 $v_1, \dots, v_n$ 的连通图)

$T :=$ 只包含顶点 v_1 的树;

visit(v_1);

Procedure visit(v : G 的顶点)

for v 每个邻居 w {

if w 不在 T 中 **then** {

 加入顶点 w 和边 $\{v, w\}$ 到 T ;

visit(w);

 }

}

回溯（八皇后）

在 $n \times n$ 格的棋盘上放置彼此不受攻击的 n 个皇后。

从空棋盘开始

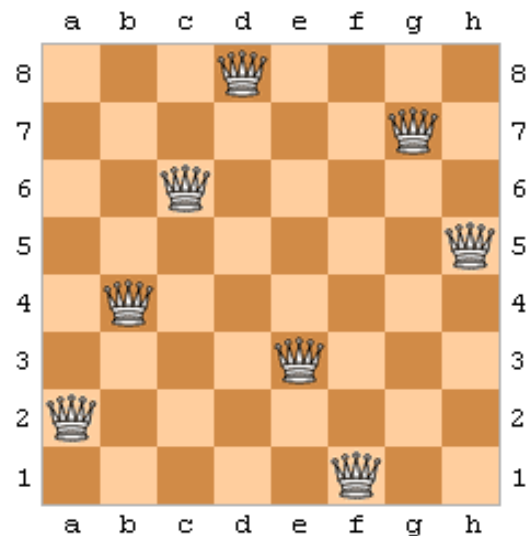
尝试第1列，第1行，... n 行；

尝试第2列，第1行，... n 行；

....

尝试第 $k+1$ 列，第1行，... n 行；

...



回溯（子集和）

给定一组正整数 $x_1, \dots, x_n$ ，和为 M 的一个子集？

从空子集开始

尝试添加一项，

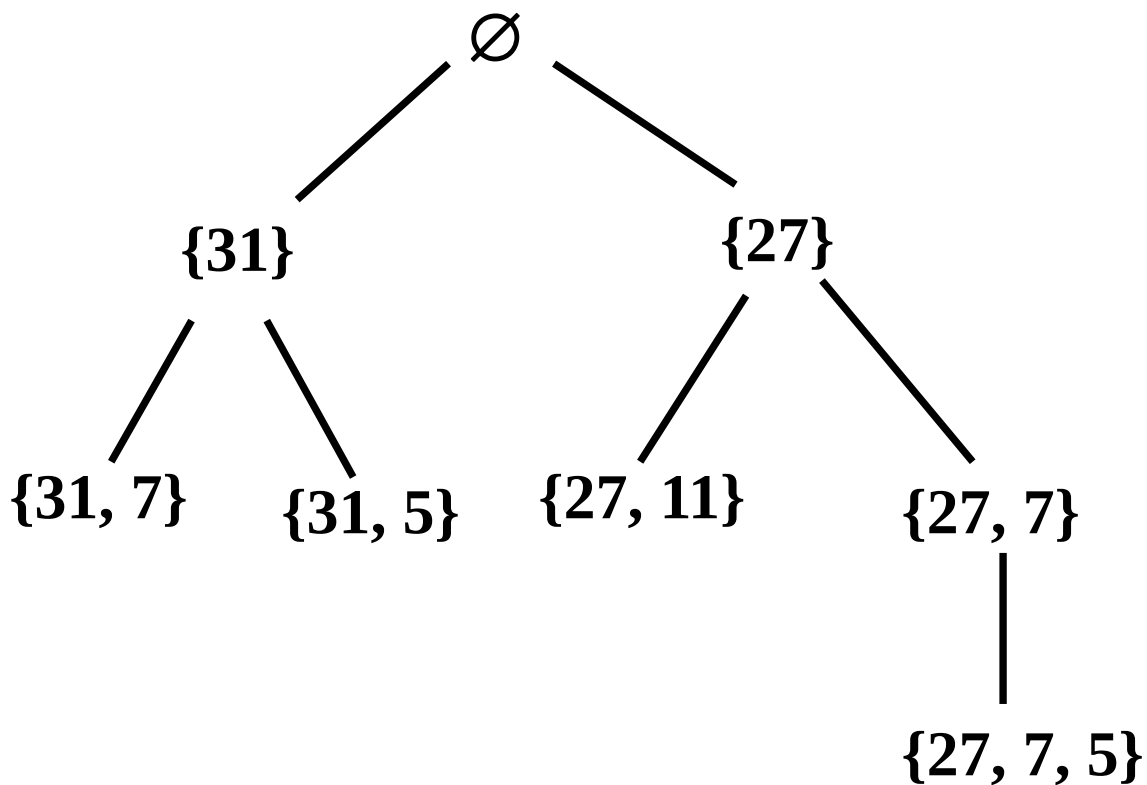
和等于 M ，结束；

和不超 M ，子集包含它；

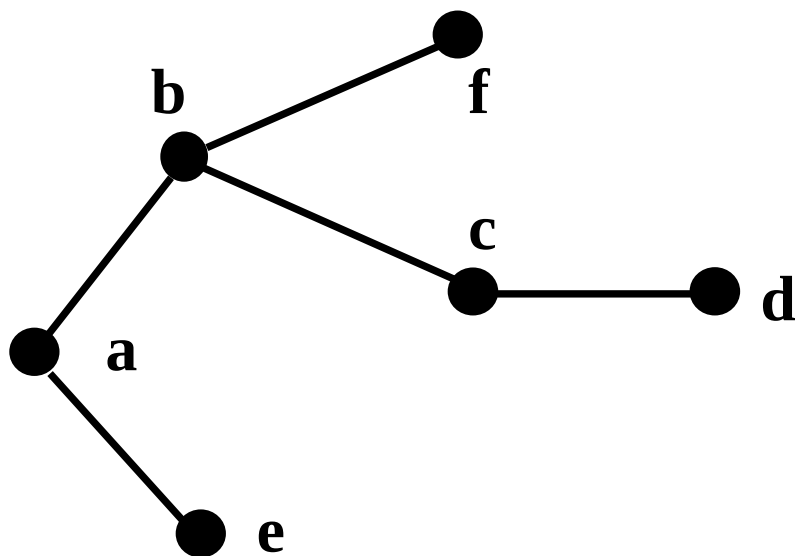
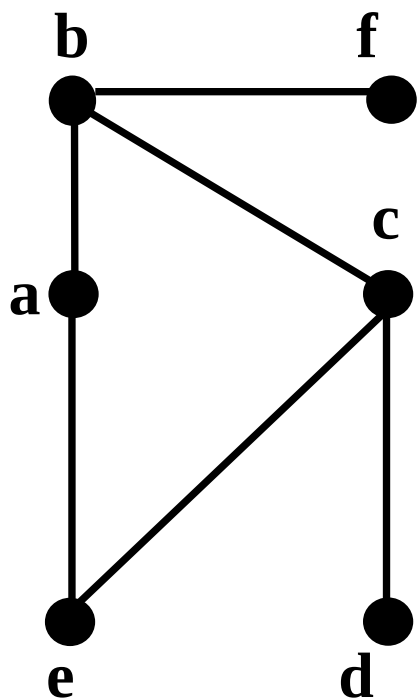
没有合适添加项，去掉最后一项，

回溯（子集和）

举例：{31, 27, 15, 11, 7, 5}, 和为39的子集？



构造生成树：广度优先搜索



广度优先搜索算法

Procedure BFS(G: 带顶点 $v_1, \dots, v_n$ 的连通图)

T:=只包含顶点 v_1 的树; **L**:=空表; 把 v_1 放入表**L**中

While **L**非空 {

 删除**L**中的第一个顶点 v ;

for v 的每个邻居 w {

if w 既不在**L**中也不在**T**中 **then** {

 加入 w 到**L**的末尾;

 加入顶点 w 和边 $\{v, w\}$ 到**T**;

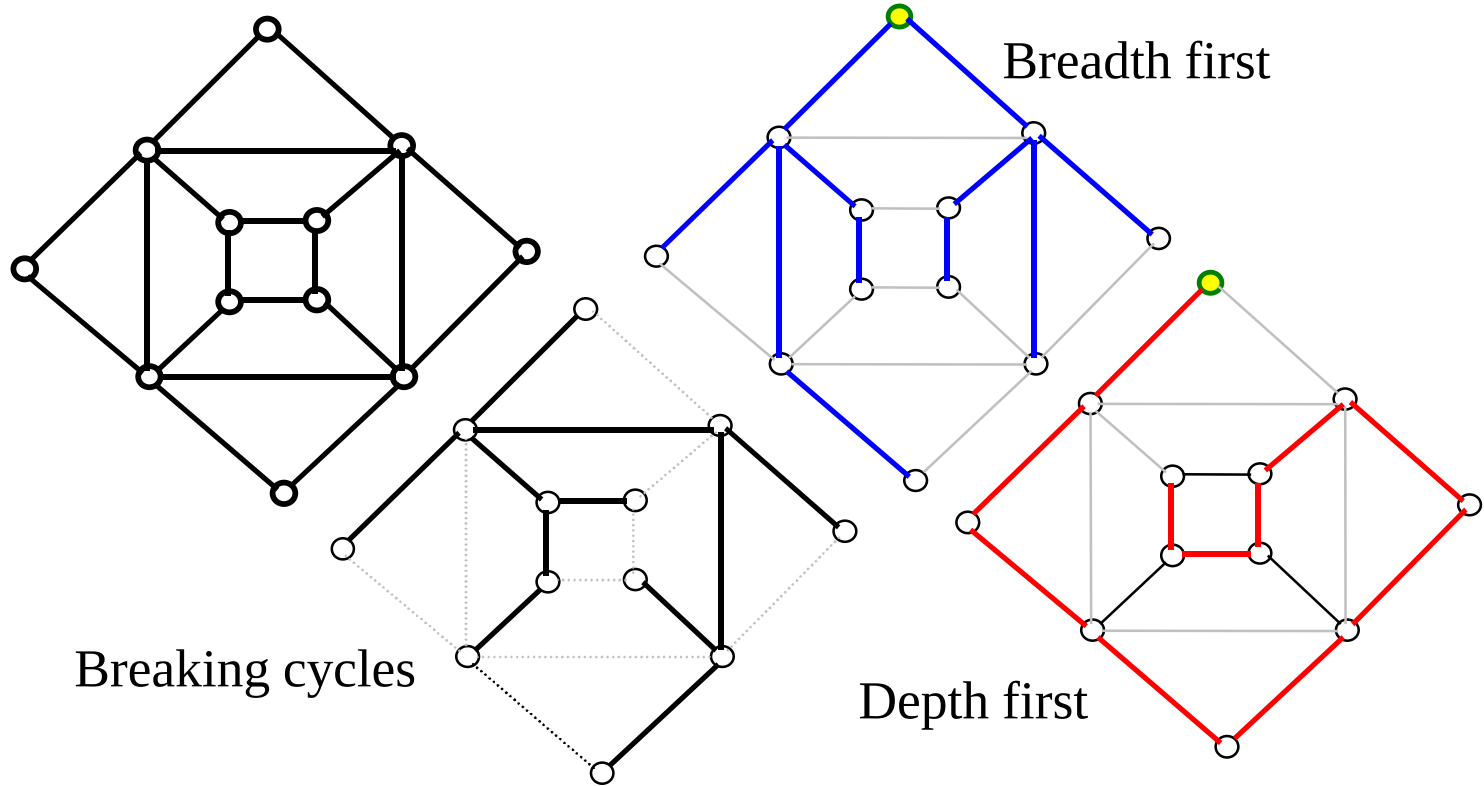
 }

 }

}

例

□ 不同的生成树



本节提要

- 内容1：生成树及其构造方法
- 内容2：最小生成树算法

最小生成树 Minimum Spanning Tree

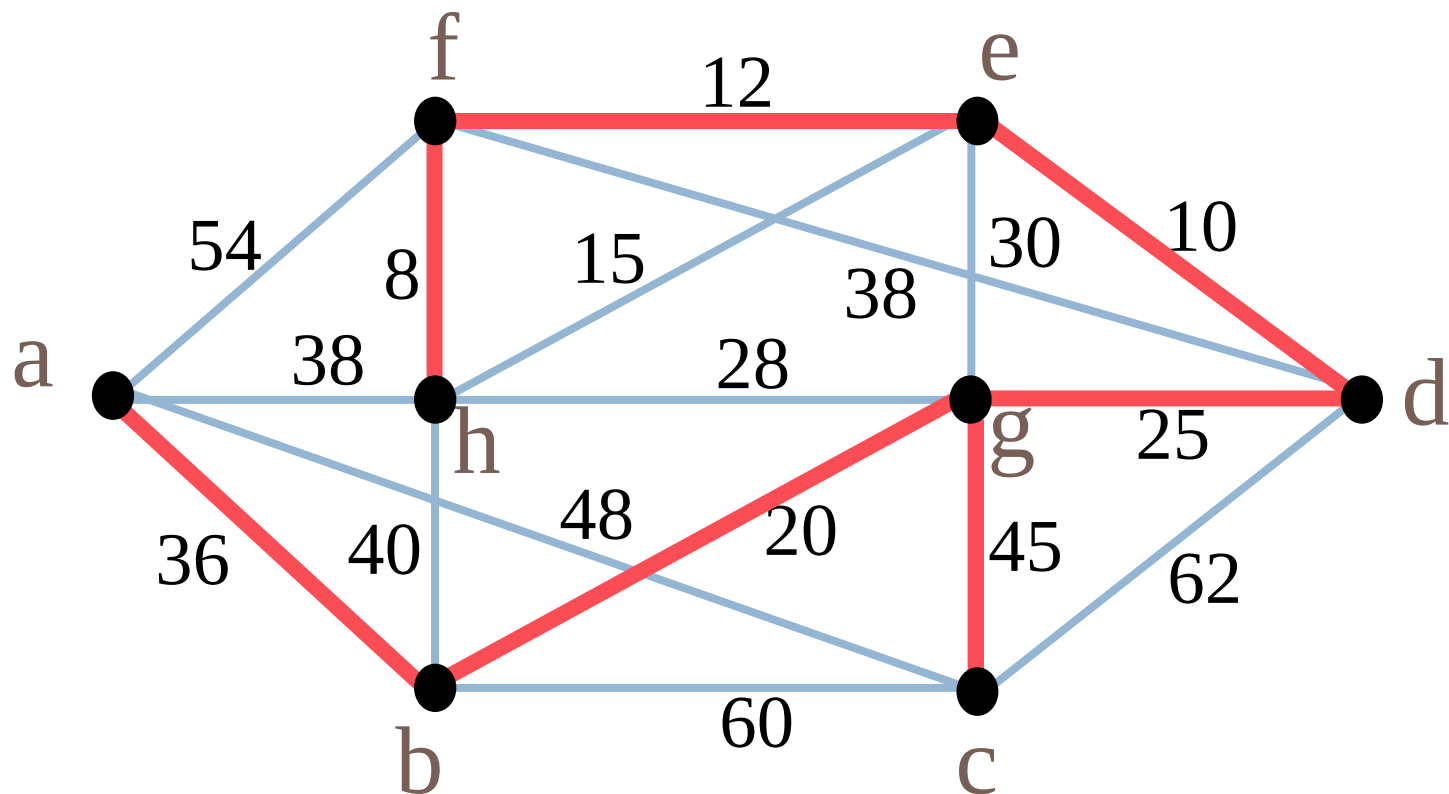
- 考虑边有权重的连通无向图。其生成树可能不唯一。定义生成树的权重为其所含各边之和。一个带权连通图的最小生成树是其权重最小的生成树。
 - ▣ 注意，这里的最小(**Minimum**)并不意味着唯一。
- 最小生成树有广泛的应用。

Prim算法（求最小生成树）

- 1: $E=\{e\}$, e 是权最小的边
 - 2: 从 E 以外选择与 E 里顶点关联, 又不会与 E 中的边构成回路的权最小的边加入 E
 - 3: 重复第2步, 直到 E 中包含 $n-1$ 条边
- 算法结束

Prim算法（举例）

- 铺设一个连接各个城市的光纤通信网络（单位：万元）。

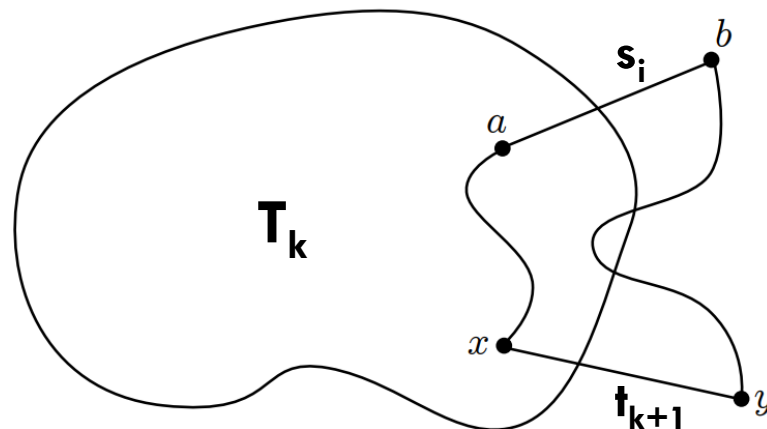


Prim 算法的正确性

令 T 为Prim算法的输出，并假设 T 按照边被选择的顺序包含了边 $\{t_1, t_2, \dots, t_{n-1}\}$ 。定义 $T_i = \{t_1, t_2, \dots, t_i\}$ 。只需证明 T_i 在一个MST中。

假设 T_k 在一个MST T' 中，如果 t_{k+1} 不属于 T' ，则 $T' + t_{k+1}$ 存在回路。

令 s_i 为回路上不在 T_k 的、且 s_i 的一个端点在 T_k 中的边。这也就意味着，当选择 t_{k+1} 时 s_i 也可选而没有被选，即 t_{k+1} 的权重不大于 s_i 的权重。那么 $T' - s_i + t_{k+1}$ 为包含 T_{k+1} 的MST。



Kruskal算法（求最小生成树）

1: $E = \{ \}$

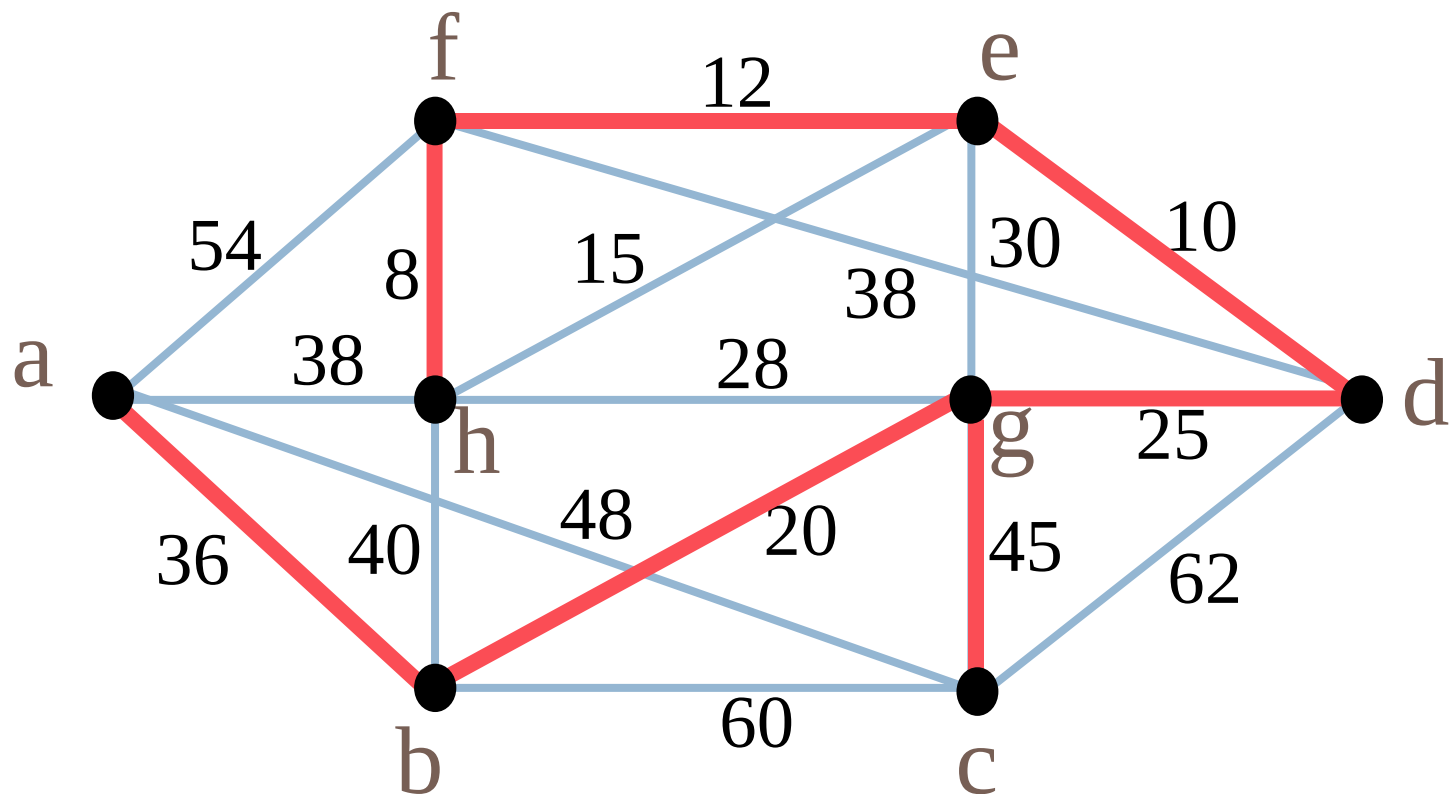
2: 从 E 以外选择不会与 E 中的边构成回路的权最小的边加入 E

3: 重复第2步，直到 E 中包含 $n-1$ 条边

算法结束

Kruskal算法（举例）

- 铺设一个连接各个城市的光纤通信网络（单位：万元）。

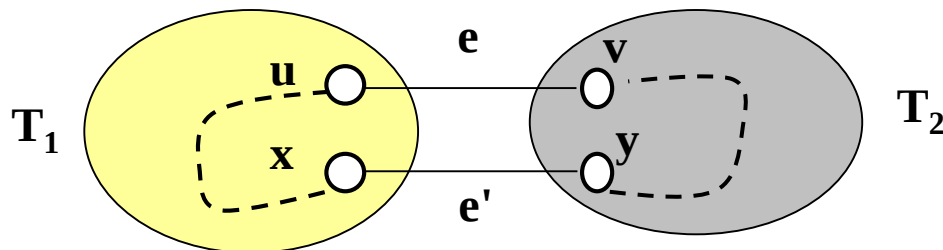


引理（更换生成树的边）

- T 与 T' 均是图 G 的生成树，若 $e \in E_T$ 且 $e \notin E_{T'}$ ，则必有 $e' \in E_{T'}$ ， $e' \notin E_T$ ，且 $T - \{e\} \cup \{e'\}$ 和 $T' - \{e'\} \cup \{e\}$ 均是 G 的生成树。

- 设 $e = uv$ ， $T - \{e\}$ 必含两个连通分支，设为 T_1, T_2 。因 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 $T - \{e\} \cup \{e'\}$ 是生成树。

而 $T' - \{e'\}$ 中 x, y 分属两个不同的连通分支，但在 $T^* = T' - \{e'\} \cup \{e\}$ 中，**xu-通路+e+vy通路** 是一条 xy -通路，因此 $T' - \{e'\} \cup \{e\}$ 连通，从而 $T' - \{e'\} \cup \{e\}$ 是生成树。



Kruskal算法的正确性

- 显然 T 是生成树。
- 按在算法中加边顺序, T 中边是 $e_1, e_2, \dots, e_{k-1}, e_k, \dots, e_{n-1}$ 。
- 假设 T 不是最小生成树。对于任意给定的一棵最小生成树 T' , 存在唯一的 k , 使得 $e_k \notin E_{T'}$, 且 $e_i \in E_{T'} (1 \leq i < k)$ 。设 T' 是这样的一棵最小生成树, 使得上述的 k 达到最大。
- 根据前述引理, T' 中存在边 e' , e' 不属于 T , 使得 $T^* = T' - \{e'\} \cup \{e_k\}$ 也是生成树。 $e' \in T'$ 与 $e_1, e_2, \dots, e_{k-1}$ 不会构成回路, 因此 $w(e') \geq w(e_k)$ 。所以 $w(T^*) \leq w(T')$, 即 T^* 也是最小生成树。但 T^* 包含 $e_1, e_2, \dots, e_{k-1}, e_k$, 矛盾。

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)

1 $A \leftarrow \emptyset$

2 **while** A does not form a spanning tree

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

4 $A \leftarrow A \cup \{(u, v)\}$

5 **return** A

Output: a minimal spanning tree of G

“避圈法”与“破圈法”

- 上述算法都是贪心地增加不构成回路的边，以求得最优树，通常称为“避圈法”；
- 从另一个角度来考虑最优树问题，在原连通带权图 G 中逐步删除构成回路中权最大的边，最后剩下的无回路的子图为最优树。我们把这种方法称为“破圈法”。

作业

- 见课程网站

例题

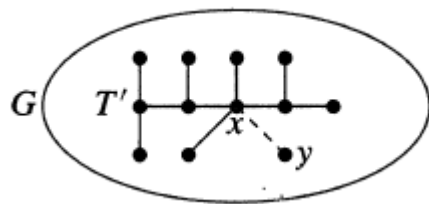
26

- 证明：如果一个连通图的边权重互不相等，则其具有唯一的**MST**。
 - 假设**G**有两棵不同的**MST** **T1**与**T2**，其边的集合按照权值从小到大排列分别是 $e_1, e_2, \dots, e_m$ 和 $e_1', e_2', \dots, e_m'$ 。假设*i*是最小的、使得 e_i 和 e_i' 不相同的边。不失一般性，假设 e_i 的权值小于 e_i' 。
 - 考察**T2**+ e_i ，必然存在包含 e_i 的回路，并且这个回路中必然包含了不同于 $e_1, e_2, \dots, e_{i-1}$ 的在**T2**中的边。假设这条边是 e' 。根据假设可知： e' 的权值大于 e_i 的权值。因此我们可以从**T2**中删除 e' 得到一棵更小的生成树。这和**T2**是**MST**树矛盾。

例题

27

- 假设树 T 有 k 条边，简单图 G 的最小度数为 k 。则 G 一定包含形如 T 的子图。
- 证明： $k=0$ 显然成立。假设 $k \leq n$ 时成立。 $k=n+1$ 时：
- 从 T 中删除一个叶子节点 v ， $T'=T-v$ 。根据归纳假设，则 G 一定包含 T' 这个子图。令 x 为 T 中与 v 相邻的点， $d_G(x) \geq k$ 且 T' 中除了 x 以外有 $k-1$ 个点，所以 x 必定和 T' 以外的某个点 y 相邻。 $T'+xy$ 边即为题意中的 T 。



例题

28

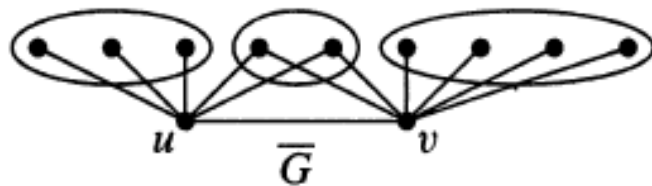
- 图的直径指图中任意两点间最短距离的最大值。
- 某个节点 u 的离心率指 u 到其他节点最短距离的最大值。
- 图的半径指所有节点的离心率的最小值。
- 显然，图直径等于所有节点离心率的最大值。

- 图的中心指所有最小离心率节点的导出子图。
- 显然，图的中心是自身当且仅当直径等于半径。

例题

29

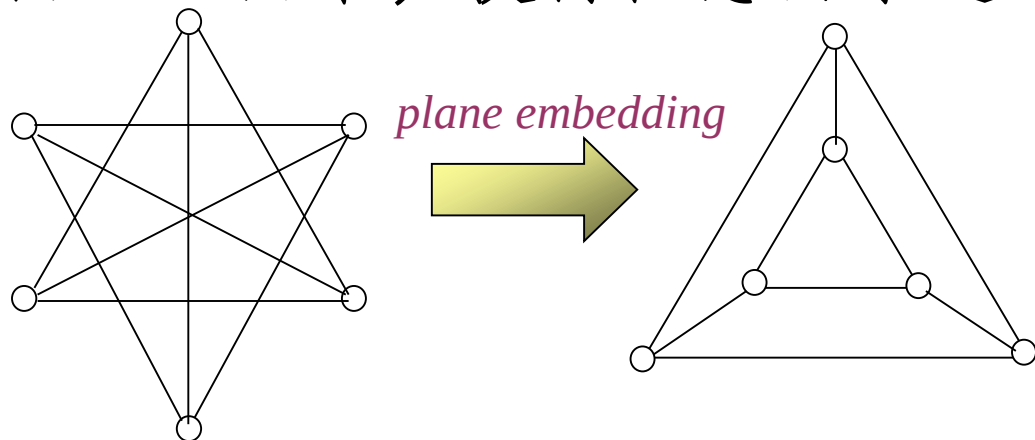
- 证明：如果简单图 G 的直径大于2，则其补图直径小于4。
- 证明： G 的直径大于2，则必定存在两个不相邻的节点 u 和 v ，它们之间没有公共邻居。即任意一个节点 x 至少跟 u 和 v 其中的一个不相邻。即任意 x 在 $\bar{G}$ 中至少跟 u 和 v 中的一个相邻。又因为 uv 边属于 $\bar{G}$ ，则 $\bar{G}$ 中任意的两点 x 和 y 通过 uv 边三跳可达。



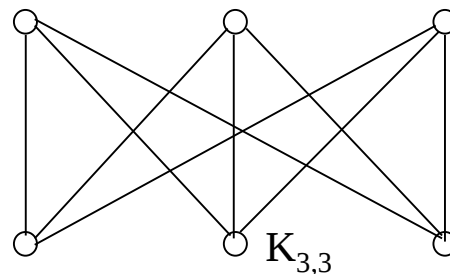
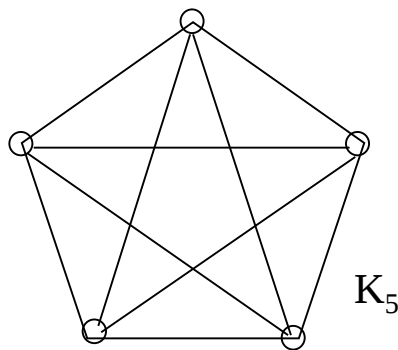
平面图

30

- 可以在一个平面上画出来，使得任意两条边不交叉

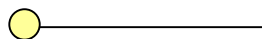
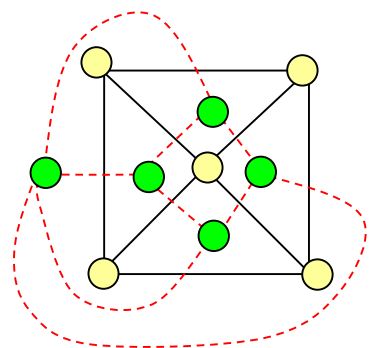


- 典型的非平面图



平面图的对偶图

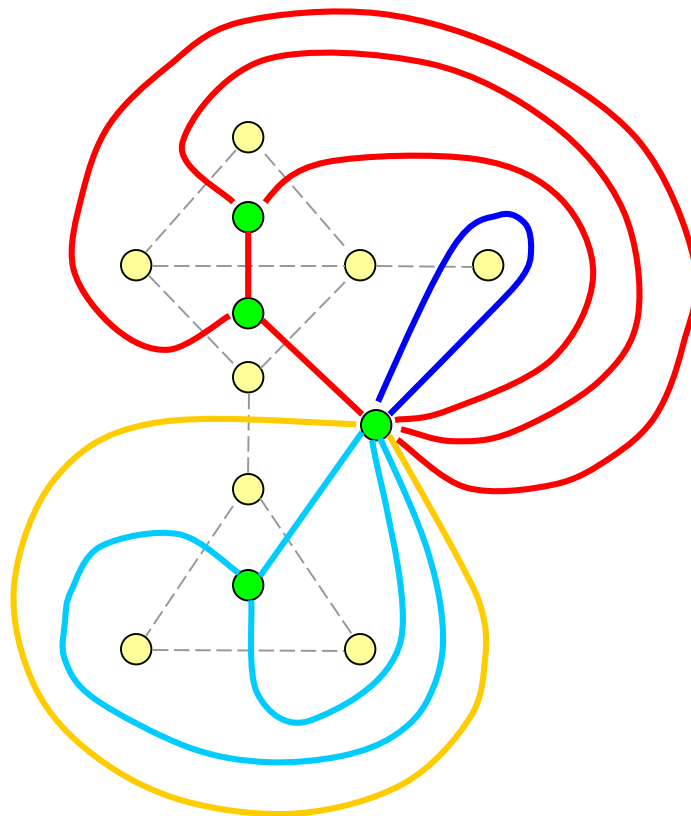
31



Vertices and edges
in G



Vertices and edges
in the dual graph G^*
of G



G 平面图充要条件: G^* 平面图

欧拉公式: $V-E+F=2$