7.1 图的定义和术语

图的结构定义:

图是由一个顶点集 V 和一个弧集 R构成的数据结构。

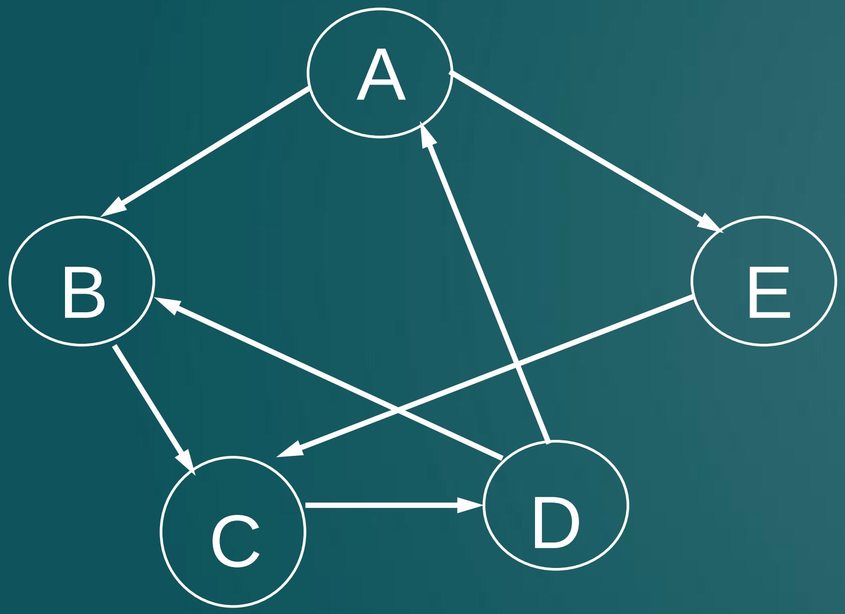
Graph = (V , R)　　R={VR}

其中，VR = {<v,w>| v,w∈V 且 P(v,w)}， <v,w>表示从 v 到 w 的一条弧，并称 v 为弧尾，w 为弧头。

谓词 P(v,w) 定义了弧 <v,w>的意义或信息。

由于“弧”是有方向的，因此称由顶点集和弧集构成的图为**有向图**。

例如: $G_1 = (V_1, VR_1)$



其中

$V_1=\{A, B, C, D, E\}$

$VR_1=\{<A,B>, <A,E>,$

$<B,C>, <C,D>, <D,B>,$

$<D,A>, <E,C> \}$

若$<v, w> \in VR$ 必有$<w, v> \in VR$，称 (v,w) 为顶点v 和顶点 w 之间存在一条**边**。

由顶点集和边集构成的图称作**无向图**。

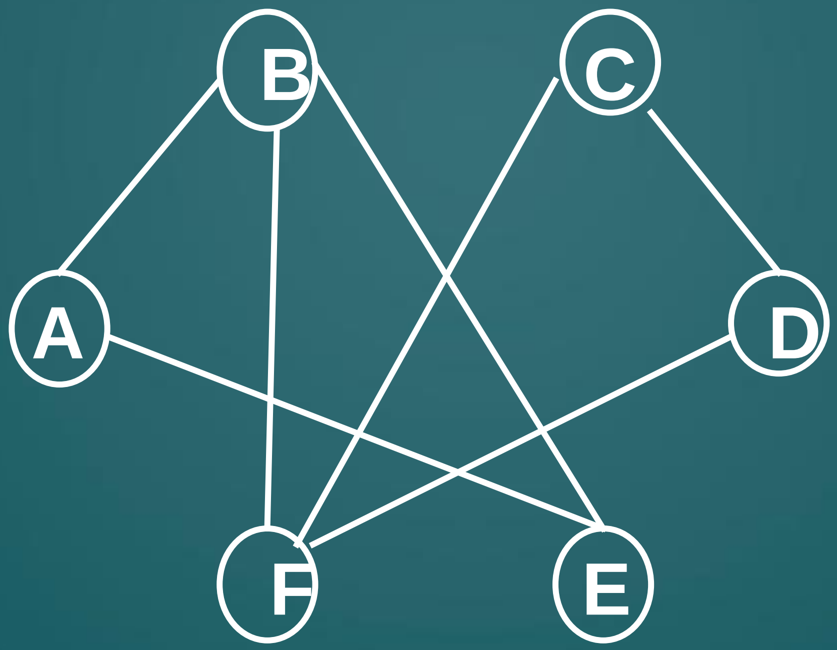
例如: $G_2=(V_2,VR_2)$

$V_2=\{A, B, C, D, E, F\}$

$VR_2=\{<A,B>, <A,E>,$

$<B,E>, <C,D>, <D,F>,$

$<B,F>, <C,F> \}$



基本操作：

结构的建立和销毁

对顶点的访问操作

插入或删除顶点

插入和删除弧

对邻接点的操作

遍历

结构的建立和销毁

CreatGraph(&G, V, VR):

// 按定义(V, VR) 构造图

DestroyGraph(&G):

// 销毁图

对顶点的访问操作

LocateVex(G, u);

// 若G中存在顶点u，则返回该顶点在

// 图中 “位置” ；否则返回其它信息。

GetVex(G, v); // 返回 v 的值。

PutVex(&G, v, value);

// 对 v 赋值value。

对邻接点的操作

FirstAdjVex(G, v);

// 返回 v 的“**第一个邻接点**” 。若该顶点

//在 G 中没有邻接点，则返回“空”。

NextAdjVex(G, v, w);

// 返回 v 的（相对于 w 的） “**下一个邻接点**”。

//若 w 是 v 的最后一个邻接点，则返回“空”。

插入或删除顶点

InsertVex(&G, v);

//在图G中增添新顶点v。

DeleteVex(&G, v);

// 删除G中顶点v及其相关的弧。

插入和删除弧

InsertArc(&G, v, w);

// 在G中增添弧<v,w>，若G是无向的，

//则还增添对称弧<w,v>。

DeleteArc(&G, v, w);

//在G中删除弧<v,w>，若G是无向的，

//则还删除对称弧<w,v>。

遍　历

DFSTraverse(G, v, Visit());

//从顶点v起**深度优先**遍历图G，并对每

//个顶点调用函数Visit一次且仅一次。

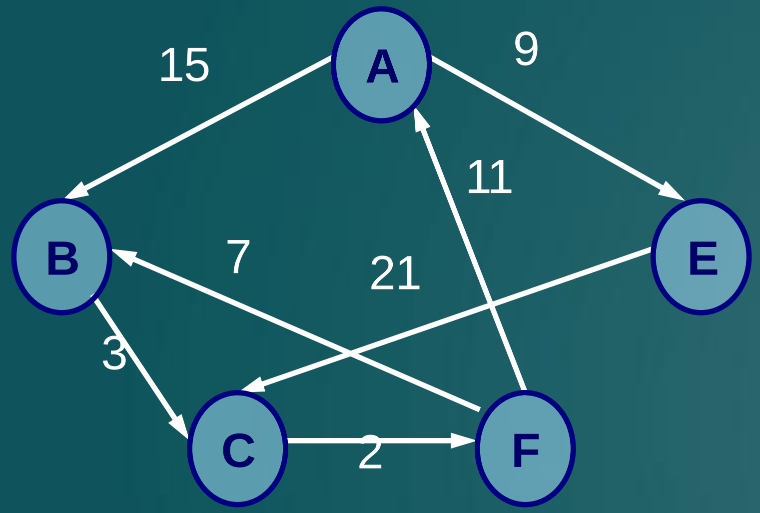
BFSTraverse(G, v, Visit());

//从顶点v起**广度优先**遍历图G，并对每

//个顶点调用函数Visit一次且仅一次。

名词和术语

网、子图



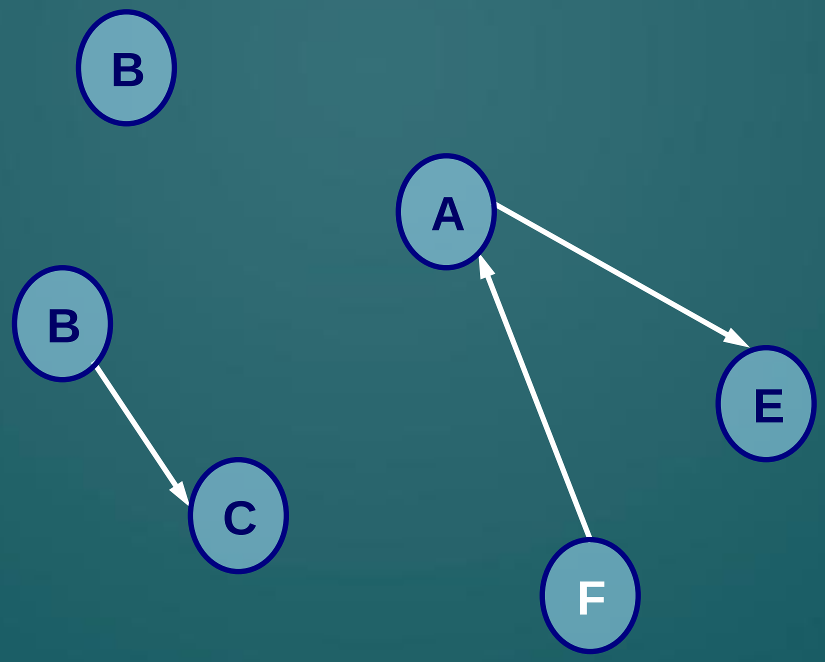
弧或边带权的图分别称作**有向网**或**无向网**。

设图G=(V,{VR}) 和图

G′=(V′,{VR′}),

且 V′⊆V, VR′⊆VR,

则称 G′为 G 的**子图**。



完全图、稀疏图、稠密图

假设图中有 **n** 个顶点，**e** 条边，则

含有 $e=n(n-1)/2$ 条边的无向图称作完全图；

含有 $e=n(n-1)$ 条弧的有向图称作 有向完全图；

若边或弧的个数 $e<nlogn$，则称作稀疏图，否则称作稠密图。

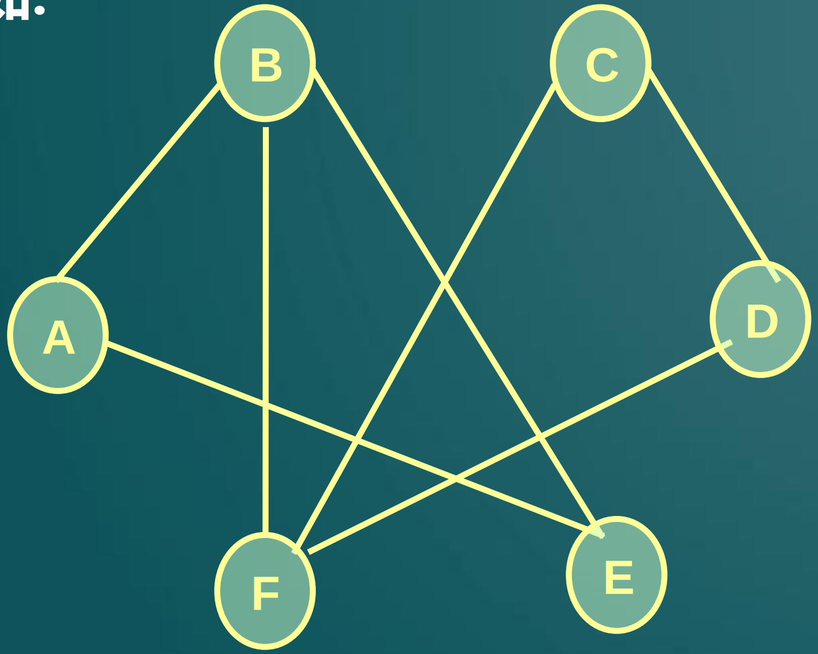
邻接点、度、入度、出度

假若顶点**v** 和顶点**w** 之间存在一条边，则称顶点**v** 和**w** 互为**邻接点**

边(**v,w**) 和顶点**v** 和**w** 相**关联**。

和顶点**v** 关联的**边的数目**定义为顶点v的**度**。记作ID

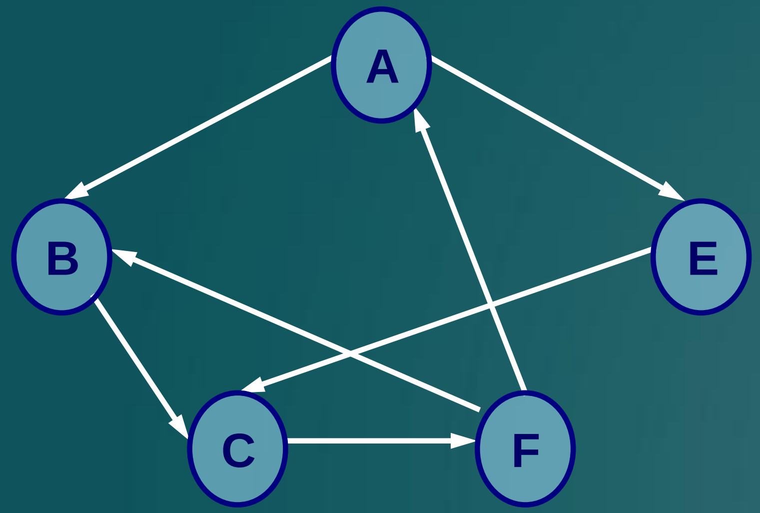
例如:



ID(B) = 3

ID(A) = 2

对有向图来说，



顶点的**出度:** 以顶点v为弧尾的弧的数目，记作OD。

顶点的**入度:** 以顶点v为弧头的弧的数目，记作ID

顶点的**度**（**TD**）=

出度（OD）+**入度**（ID）

例如:

OD(B) = 1

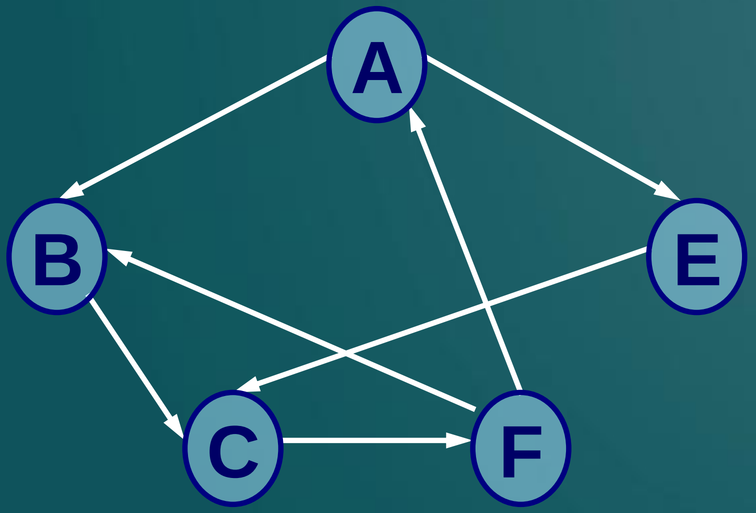
ID(B) = 2

TD(B) = 3

路径、路径长度、简单路径、简单回路

设图G=(V,{VR})中的一个顶点序列{ $u=v_{i,0}, v_{i,1}, \ldots, v_{i,m}=w$}中，$(v_{i,j-1}, v_{i,j}) \in VR$， $1 \le j \le m$，则称从顶点u 到顶点w 之间存在一条**路径**。路径上边的数目称作**路径长度**。

如:长度为3的路径{A,B,C,F}

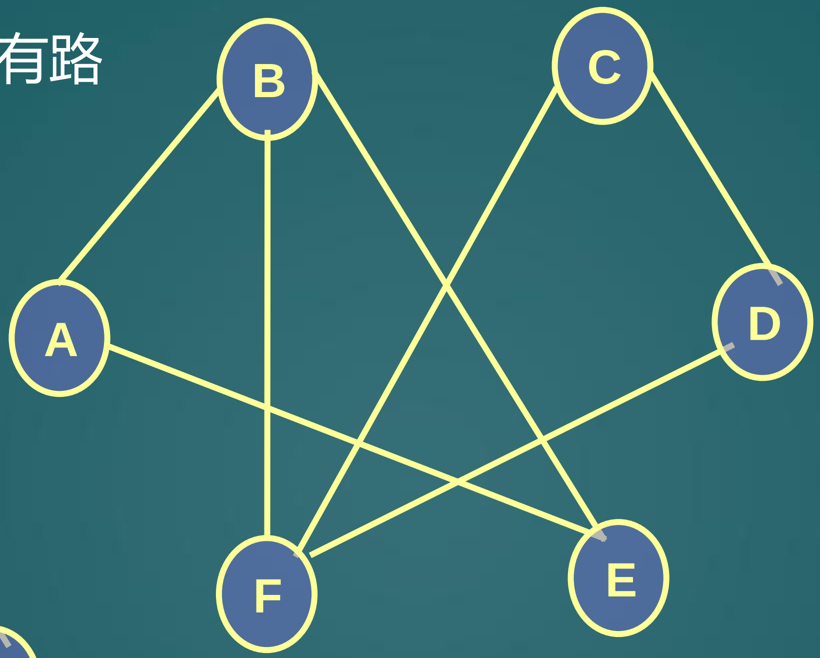


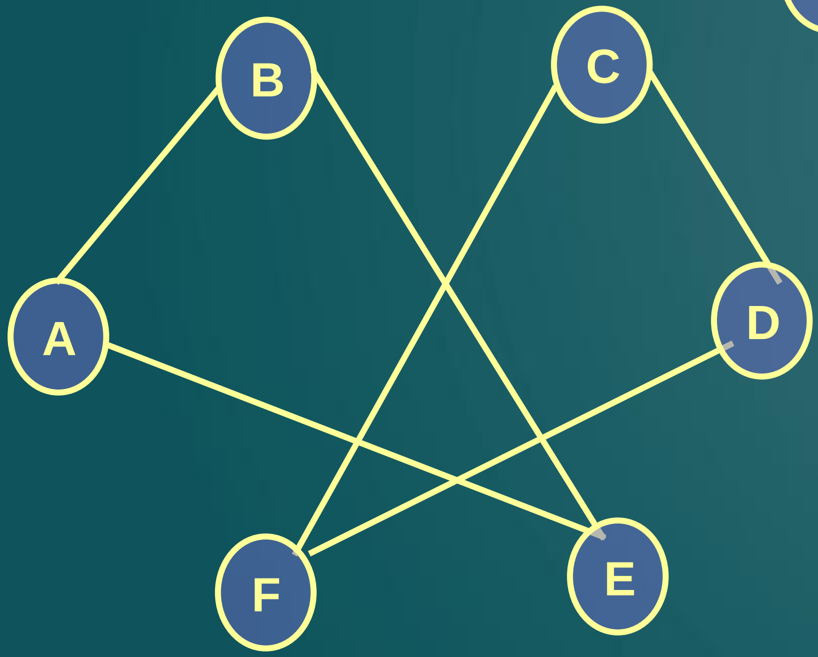
简单路径：序列中顶点不重复出现的路径。

简单回路：序列中第一个顶点和最后一个顶点相同的路径。

连通图、连通分量、强连通图、强连通分量

若图G中任意两个顶点之间都有路径相通，则称此图为**连通图**；

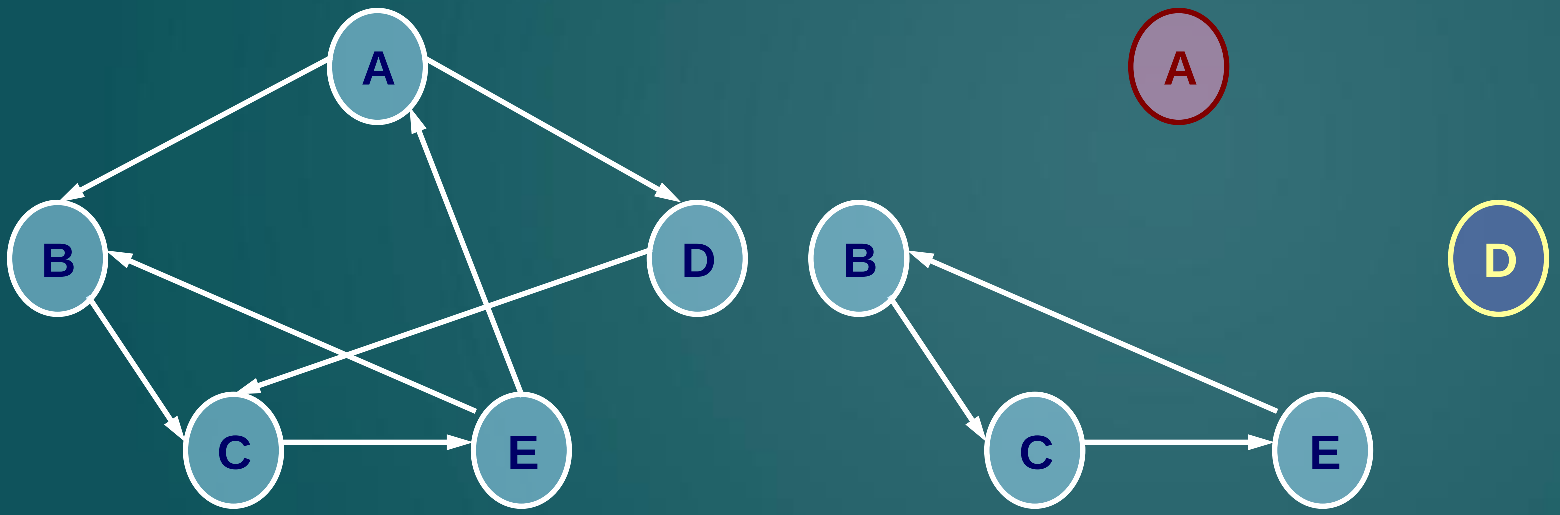




若无向图为非连通图，则图中各个极大连通子图称作此图的**连通分量**。

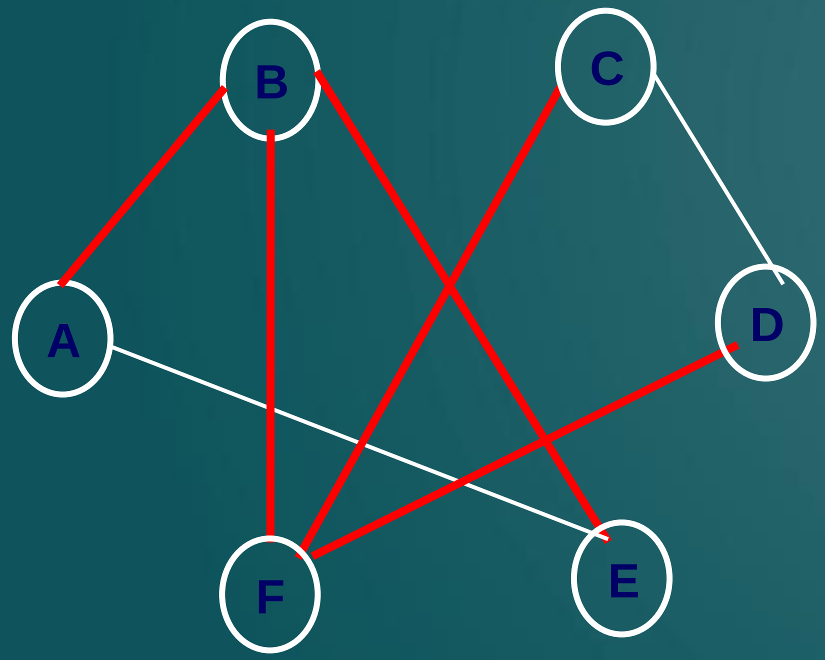
对有向图，若任意两个顶点之间都存在一条有向路径，则称此有向图为**强连通图**。

否则，其各个强连通子图称作它的**强连通分量**。



生成树、生成森林

假设一个连通图有 n 个顶点和 e 条边，其中 n-1 条边和 n 个顶点构成一个极小连通子图，称该极小连通子图为此连通图的**生成树**。



对非连通图，则称由各个连通分量的生成树构成的集合为此非连通图的**生成森林**。