**20181114**

**第6章 图**

**🕮** 6.1 概述

图(graph)定义为G = (V; E)。V中元素称为顶点(vertex)；E中元素分别对应V中某一对顶点(u, v)，表示存在某种关系，称为边(edge)。

顶点总数n = |V|；边总数：e = |E|

列表结果和树结构都是图的特例。

**🏵 6.1.1 无向图和有向图**

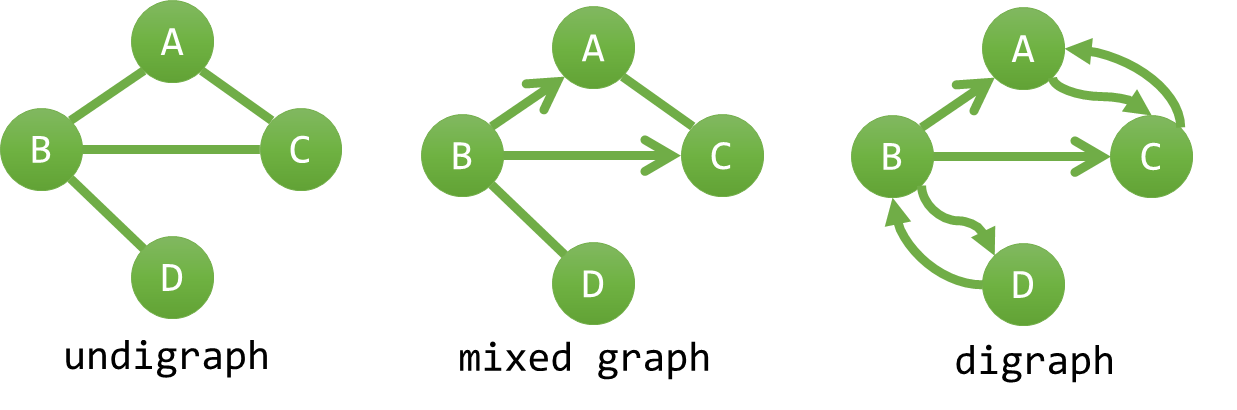
若边(u, v)对应顶点u和v的次序无所谓，则称为无向边(undirected edge)。反之若u和v不对等，则(u, v)称为有向边(directed edge)，也称弧(arc)。

无向边(u, v)与(v, u)等价；但有向边则不同。有向边(u, v)从u指向v，u称为该边的起点(origin)或尾顶点(tail)，v称为该边的终点(destination)或头顶点(head)。

所有边都是无方向的图称为无向图(undirected graph, 简称undigraph)。

所有边都是有向边的图称为有向图(directed graph, 简称digraph)。

若同时包含无向边和有向边，称为混合图(mixed graph)



有向图通用性更强，因为无向图和混合图都可转为有向图：每条无向边(u, v)都可等效地替换为一对有向边(u, v)和(v, u)。

**🏵 6.1.2 度**

对于任何边e = (u, v)，称顶点u和v彼此邻接(adjacent)，互为邻居；u和v都与边e彼此关联(incident)。

无向图中，与顶点v关联的边数称为v的度数(degree)，记为deg(v)。

对于有向边e = (u, v)，e称为u的出边(outgoing edge)、v的入边(incoming edge)。v的出边总数称为出度(out-degree)，记为outdeg(v)；入边总数称为入度(in-degree)，记为indeg(v)。

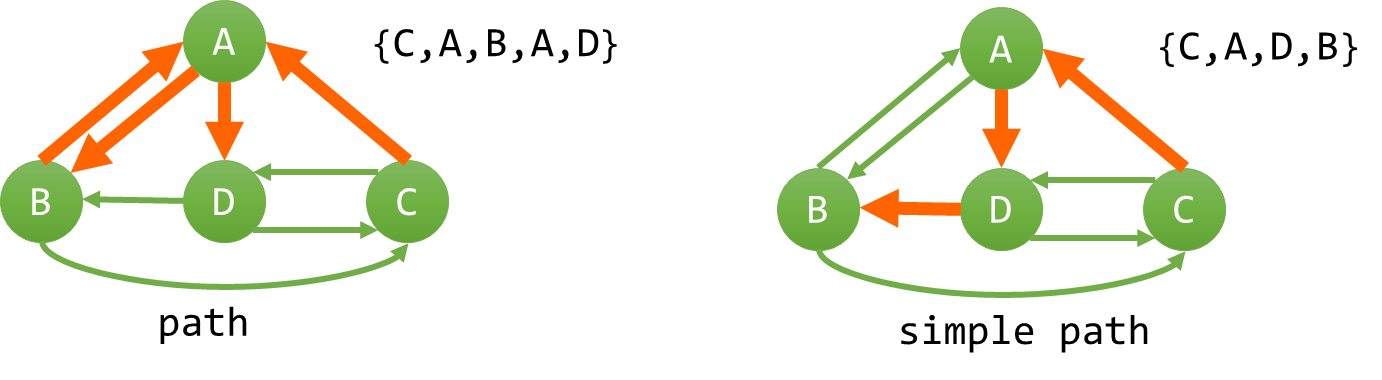
**🏵 6.1.3 通路与环路**

路径或通路(path)，是由m+1个顶点和m条边交替而成的序列：

任意一条边，路径沿途边总数为m，称为路径长度，记为

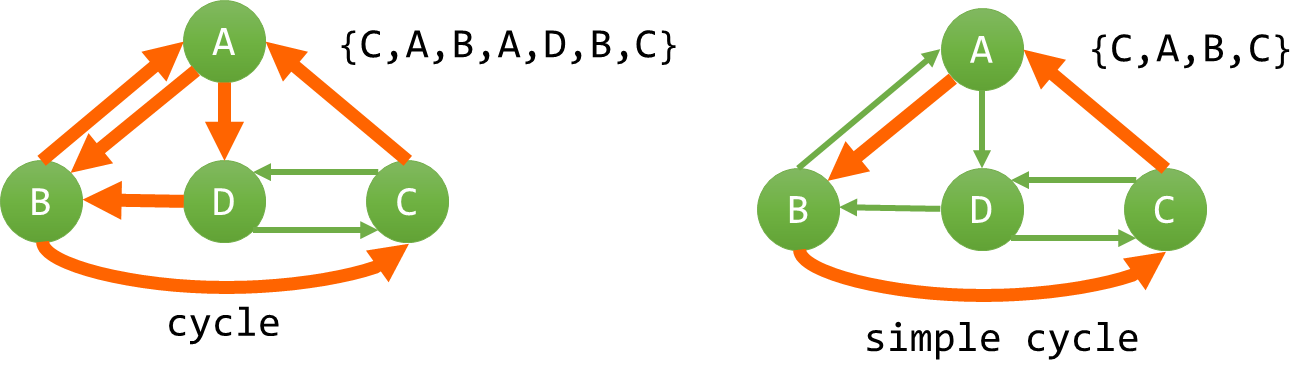
为了简化，也可省略连接其间的边，表示为：

如图左边的{C, A, B, A, D}是顶点C到D的一条通路，长度为4。虽然通路上边必须互异，但顶点却可以重复。沿途顶点互异的通路称为简单通路(simple path)。右边的{C, A, D, B}是从C到B的一条简单通路，长度为3。



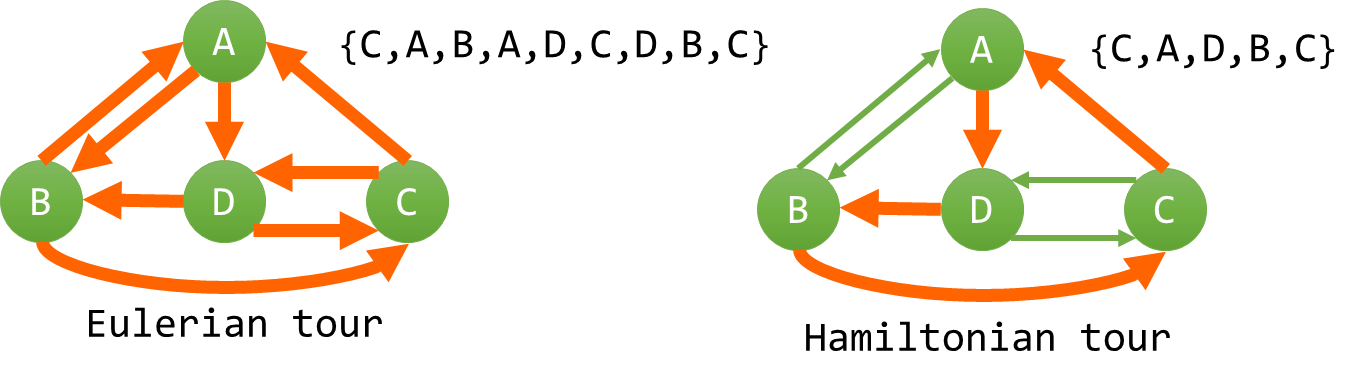
对于长度的通路，若起止顶点相同()，则称为环路(cycle)。

左边{C, A, B, A, D, B, C}是一条环路，长度为6。同样，环路沿途除外所有顶点都互异，则称为简单环路(simple cycle)，如图右边{C, A, B, C}。



特别地，经过图中所有边恰好一次的环路，称为欧拉环路(Eulerian tour)，长度等于边总数e。如图左边的{C, A, B, A, D, C, D, B, C}，长度为8。

经过图中各顶点恰好一次的环路，称为哈密尔顿环路(Hamiltonian tour)，长度为构成环路的边数，即顶点总数n。如图右边的{C, A, D, B, C}，长度为4。



**🏵 6.1.4 带权网络**

通过一个权值函数，为每一边e指定一个权重(weight)，比如wt(e)为边e的权重。各边均带有权重的图，称作带权图(weighted graph)或带权网络(weighted network)，有时也简称网络(network)，记作G(V, E, wt())。

**🏵 6.1.5 复杂度**

问题的输入规模，应以顶点数和边数总和(n+e)度量。

包含n个顶点的图，最多含有多少条边?

无向完全图：无向图中如果任意两个顶点之间都存在边，称为无向完全图；含有n个顶点的无向完全图有n(n-1)/2条边。

有向完全图：有向图中如果任意两个顶点之前都存在方向互为相反的两条边，称为有向完全图；含有n个顶点的有向完全图有n(n-1)条边。

稀疏图和稠密图：稀疏和稠密都是相对而言，通常认为边小于为稀疏图， 反之为稠密图。

因此不管有向图还是无向图，边数都不大于完全图，即：

**🏵 6.1.6 连通图**

连通(connected)：

如果无向图中存在u到v的路径，则u到v连通(v到u也连通)，因此u和v之间连通；默认任意顶点与自己连通；

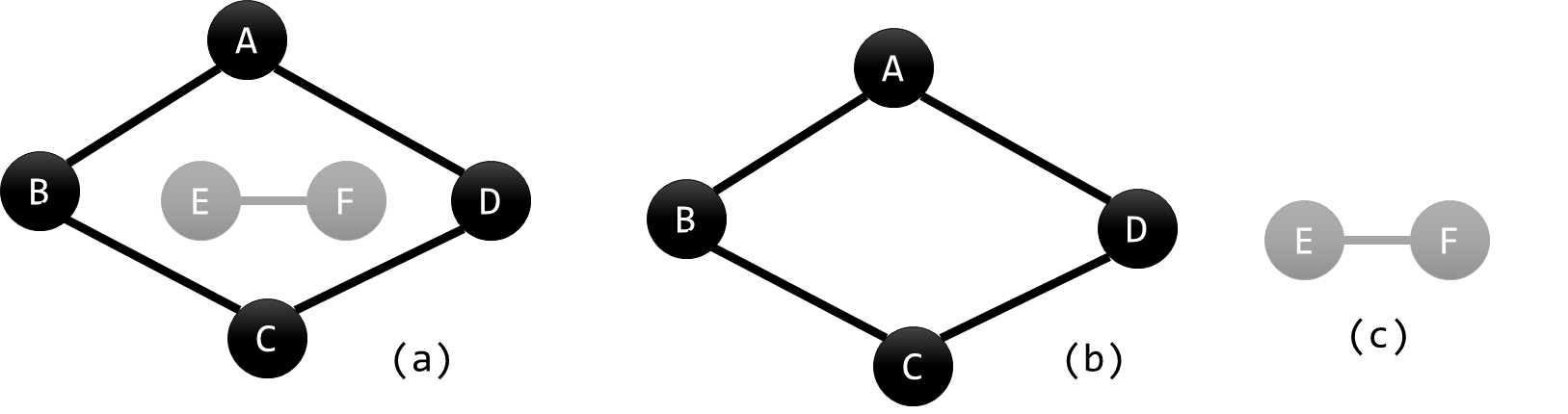
如果有向图存在u到v的路径，则u到v连通，但v到u不一定连通。

连通图(connected graph)：**无向图**G任意两个顶点u和v都是连通的。

连通分量(connected component)：无向图中极大连通子图。

**注**：1) 子图要连通；2) 连通子图含有极大顶点数；3) 具有极大顶点数的连通子图包含依附于这些顶点的所有边。

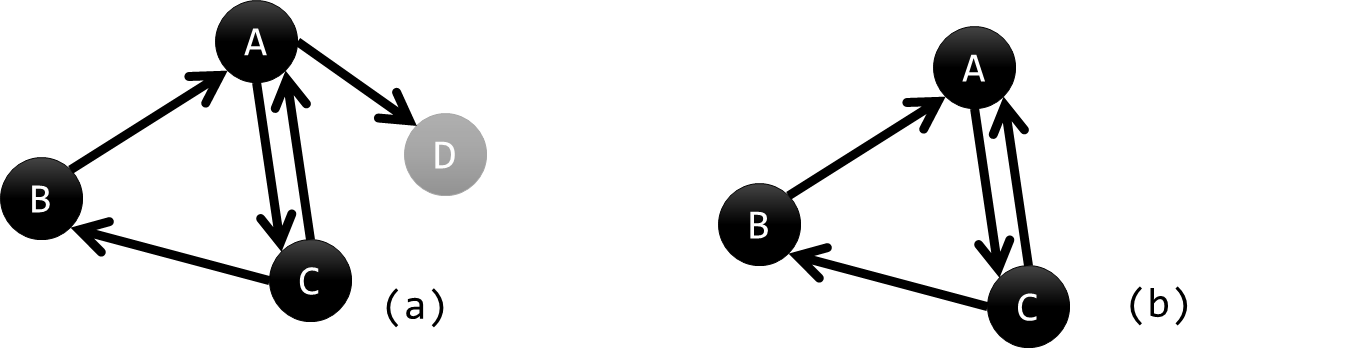
如图(a)是无向非连通图，它有两个连通分量，即(b)和(c)：



强连通图(strongly connected graph)：**有向图**G任意两个顶点u到v和v到u都连通(存在路径)。

强连通分量(strongly connected component)：有向图中极大强连通子图。

如图(a)是有向图，但不是强连通图，因为A到D连通，但D到A不连通；(b)是强连通图，而且是(a)的强连通分量。



**🕮** 6.2 **邻接矩阵**

邻接矩阵(adjacency matrix)是图ADT最基本的实现方式，使用方阵A[n][n]表示由n个顶点构成的图，其中每个单元，描述一对顶点之间可能存在的邻接关系。

顶点集采用一维向量存储，边集采用二维向量存储。

对于无权图，顶点u到v的边存在，A[u][v] = 1；反之A[u][v] = 0。对于带权网络，矩阵各单元可从布尔型改为整型或浮点型，记录对应边的权重。对于不存在的边，。

如图分别是无向图、有向图、有向带权图的邻接矩阵：

