中国科学技术大学
University of Science and Technology of China

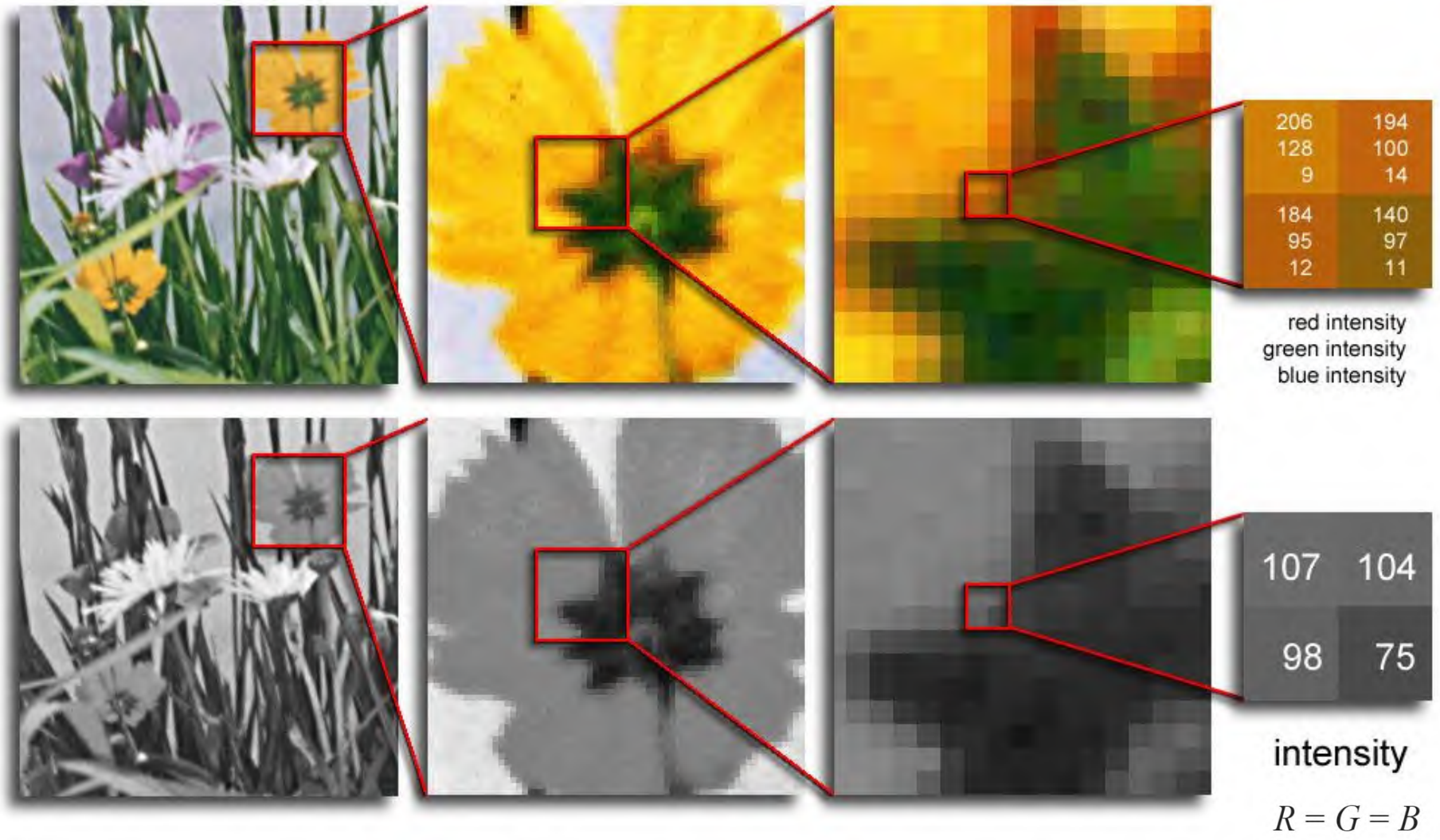
# 数学建模
# Mathematical Modeling

陈仁杰
中国科学技术大学

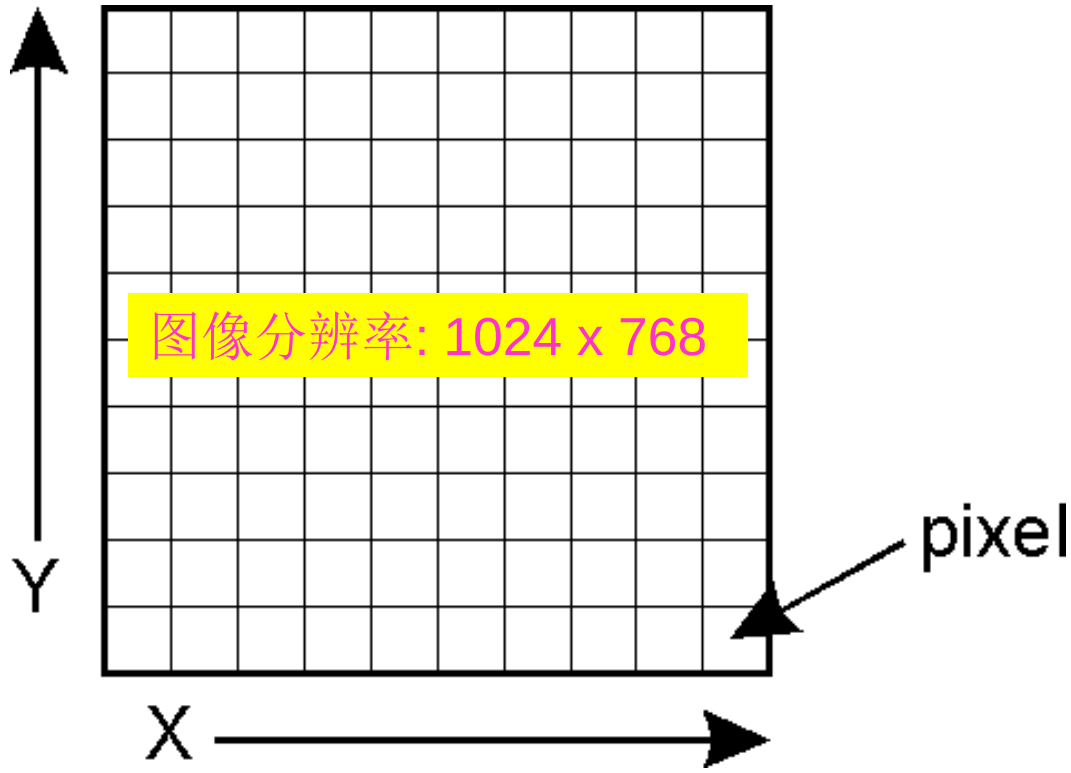
# 建模第一步：抽象

- 抽象
  - 简化非本质的因素
  - 透过现象看本质
- 迭代
  - 将问题转化为已有问题
  - 组合、递归
- 方法论：20-80原则
  - 分治法Divide-and-conquer
  - Coarse-to-fine

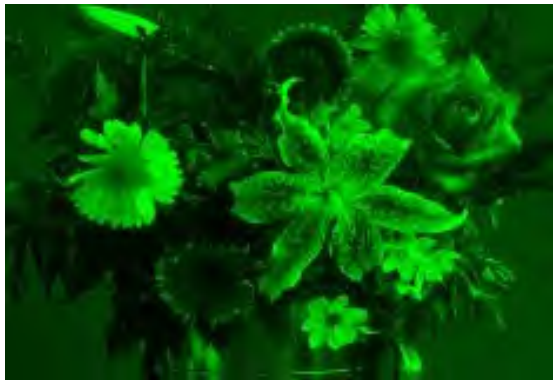
# 数字图像的抽象

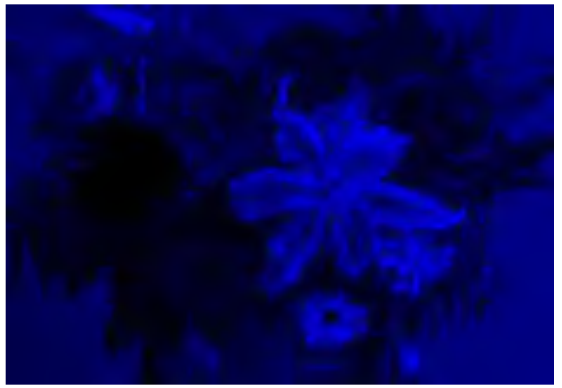

206 194
128 100
9 14
184 140
95 97
12 11
red intensity
green intensity
blue intensity
107 104
98 75
intensity
$R = G = B$

# 图像的数学表达



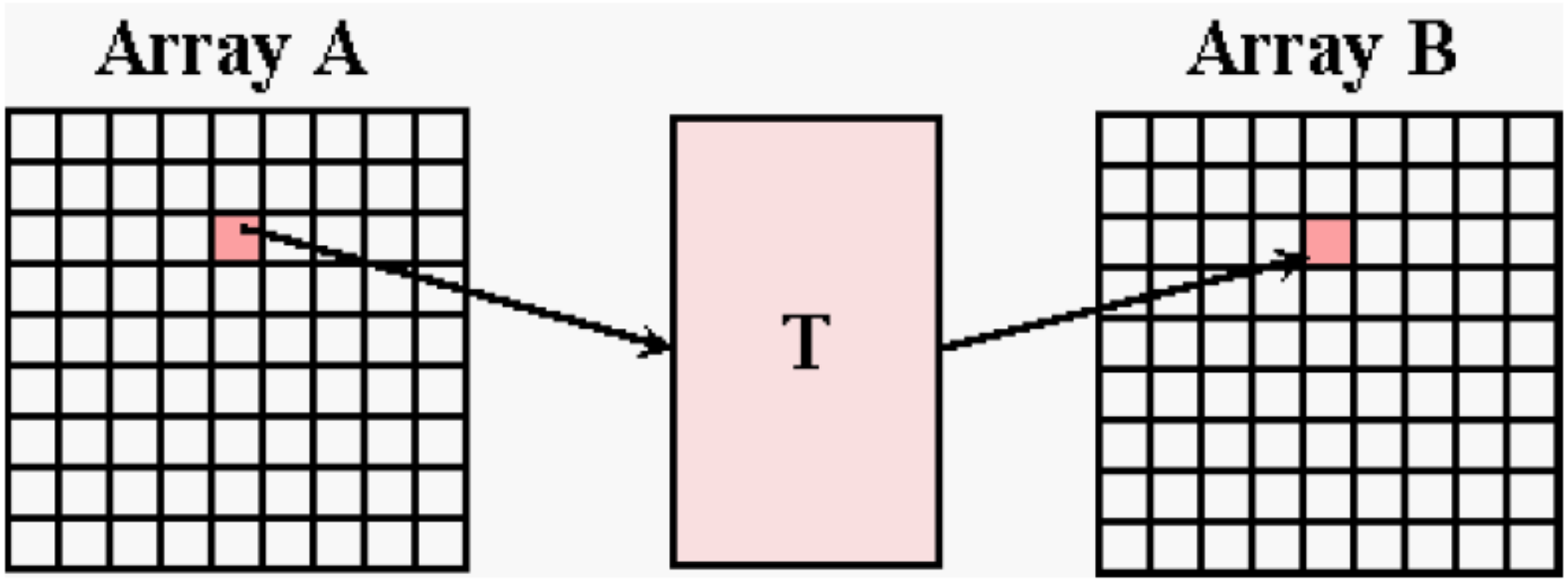
# 彩色图像的3个通道





# 图像的颜色变换

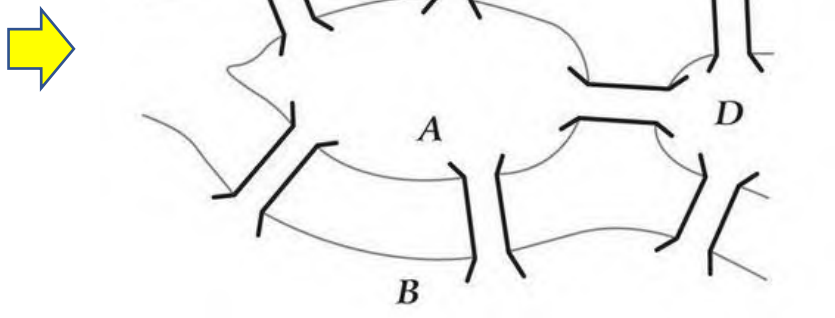
$$\mathbf{B}[x, y] = T\big[\mathbf{A}[x, y]\big]$$



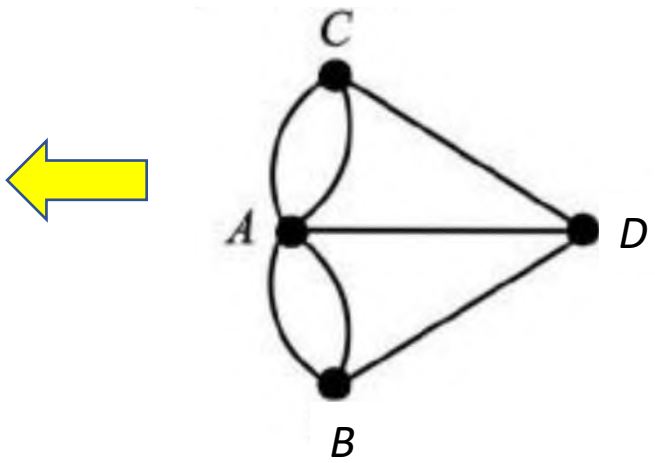
# 图与网络模型

# 欧拉七桥问题


休憩處
休憩處
休憩處
休憩處


C
A
D
B

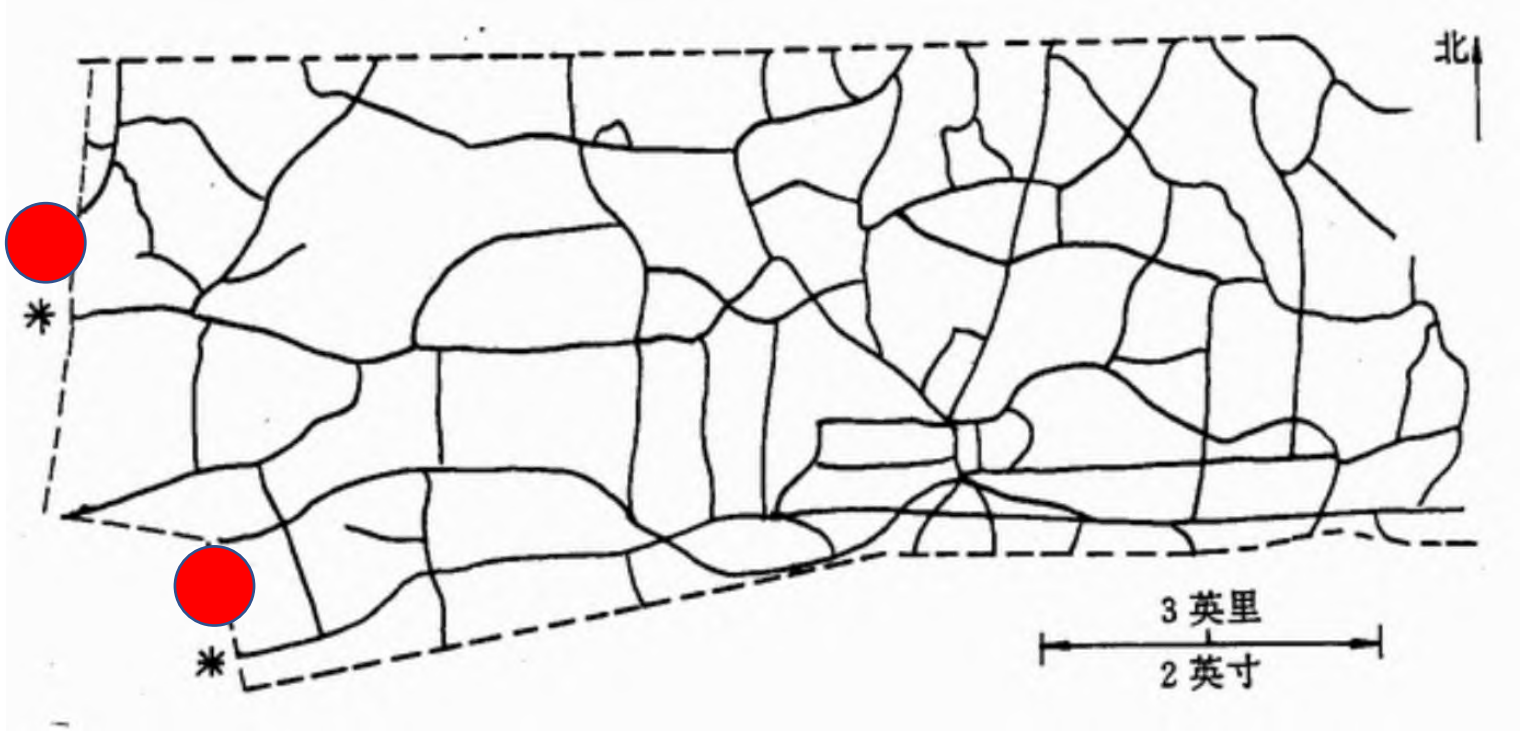

安阳市
鹤壁市
濮阳市
新乡市
开封市
郑州市
菏泽市
济宁市
泰安市
莱芜市
枣庄市
商丘市


C
A
D
B

# 问题1：扫雪问题

课本第十三章（MCM1990B）

- 安排两辆扫雪车分别从指定位置(红点处)出发清除路面上的积雪。给出清扫积雪的最佳方案。



【模型建立】

- 假设扫雪车扫雪时的行驶速度$u$，不扫雪时的行驶速度$v \gg u$。
- 假设扫雪车可在州高速公路上快速行驶，无需扫雪，速度$w > v$。
- 以道路交叉口为顶点、县道为边建立无向图$G = (V, E)$。
- 所有位于高速公路上的顶点合并为1个顶点。
- 任意两个相邻顶点之间恰有2条边。
- 设每条边$e \in E$的长度为$L(e)$。

【数学问题】

- 求$G$上两条指定起点的路$P_1, P_2$，使得$P_1 \cup P_2 \supset E$，并且

$$T = \max\left(\sum_{e \in P_1} L(e), \sum_{e \in P_2} L(e)\right)$$

尽可能小。

在一个有向图上找到使行驶路程最短的最佳方案

【问题求解】

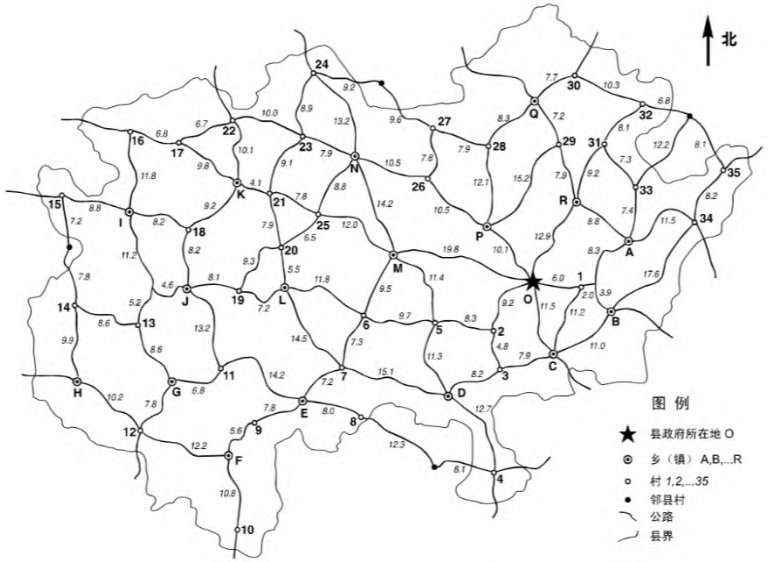
- 把$G$分解成一些两两无公共边的圈。
- 把这些圈分成长度之和近似相等的两部分，每部分是$G$的连通子图。
- 若某部分不连通，可尝试枚举调整1个圈、2个圈、…。
- 把每部分的圈拼接成一个回路。

# 问题2：灾情巡视路线

课本第十三章思考题1（CUMCM1998B）

下图为某县的乡镇村公路网示意图，公路边的数字为该路段的公里数。今年夏天该县遭受水灾。为考察灾情、组织自救，县领导决定，带领有关部门负责人到全县各乡镇村巡视。巡视路线指从县政府所在地出发，走遍各乡镇村，又回到县政府所在地的路线。

1. 若分三组（路）巡视，试设计总路程最短且各组尽可能均衡的巡视路线。
2. 假定巡视人员在各乡（镇）停留时间$T=2$小时，在各村停留时间$t=1$小时，汽车行驶速度$V=35$公里/小时。要在24小时内完成巡视，至少应分几组；给出这种分组下你认为最佳的巡视路线。
3. 在上述关于$T,t$和$V$的假定下，如果巡视人员足够多，完成巡视的最短时间是多少；给出在这种最短时间完成巡视的要求下，你认为最佳的巡视路线。
4. 若巡视组数已定（比如三组），要求尽快完成巡视，讨论$T,t$和$V$改变对最佳巡视路线的影响。



【模型建立】

- 以乡(镇)、村为顶点，连接相邻乡(镇)、村的道路为边建立无向图$G=(V,E)$。
- 计算任意两个顶点$u,v$之间的最短路长度$d(u,v)$。
- 每组巡视线路是图$G$上从顶点$O$出发的回路。

【思路】

- 把$V\backslash\{O\}$中所有顶点分成互不相交的三组$X_1,X_2,X_3$，并且最小化

$$f(X_1)+f(X_2)+f(X_3),$$

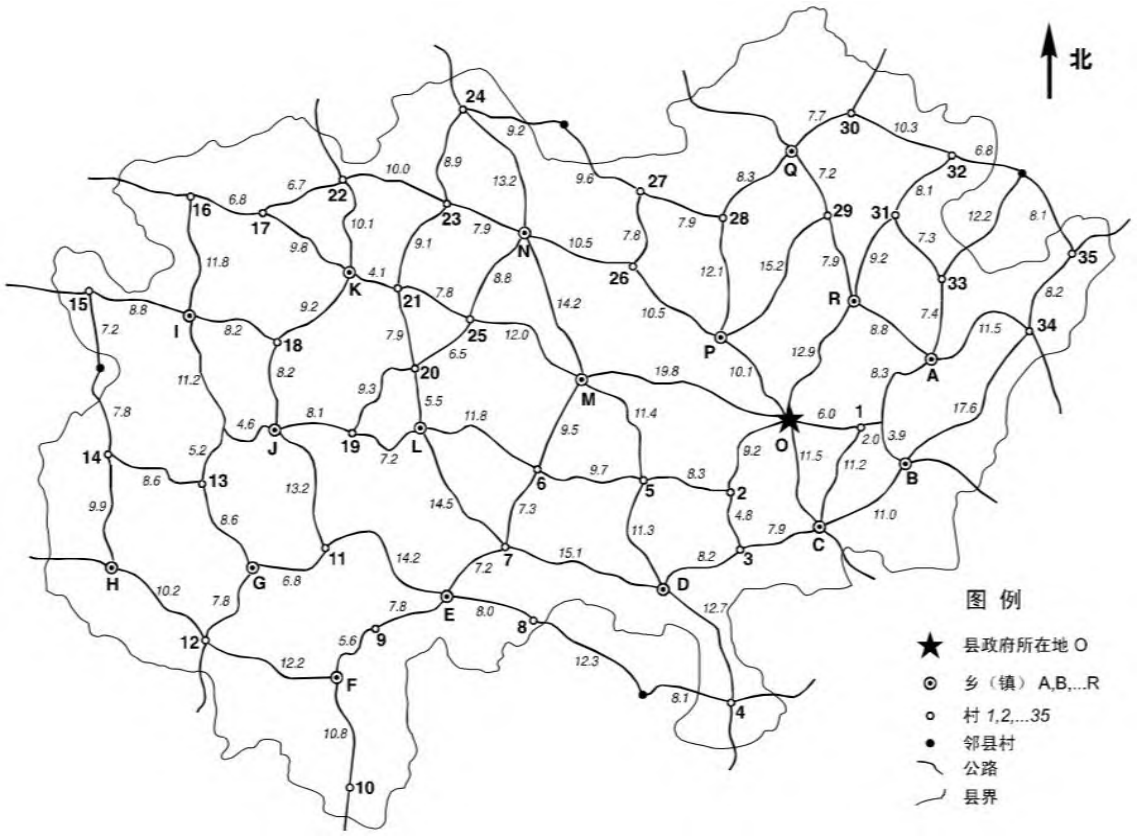
其中$f(X_i)$是经过$X_i\cup\{O\}$中所有顶点的最短回路长度。

- 使用下述算法求以上问题的局部极小值解。
- 初始时，$X_1=X_2=X_3=\emptyset$，$S=V-\{O\}$。
- 每次把某个$s\in S$移至某个$X_i$，使得$f(X_i\cup\{s\})$最小，直至$S=\emptyset$。
- 设$f(X_i)<f(X_j)$，尝试把$X_j$中的点移至$X_i$。

问题2与问题1的目标函数不同，其它相同

# 什么是图/网络?

- 离散点集及其相互关联的“抽象”
- 学科：图论
- 例子:
  - 地图
  - 互联网/物联网
  - 人际网络
  - 社交网络
  - 论文引用
  - ...



# 图论简介

# 图的定义

- 图是由顶点集合(vertex)及顶点间的关系集合组成的一种数据结构：

  Graph = ( $V$, $E$ )

其中$V = \{ x \mid x \in$ 某个数据对象$\}$ 是顶点的有穷非空集合；

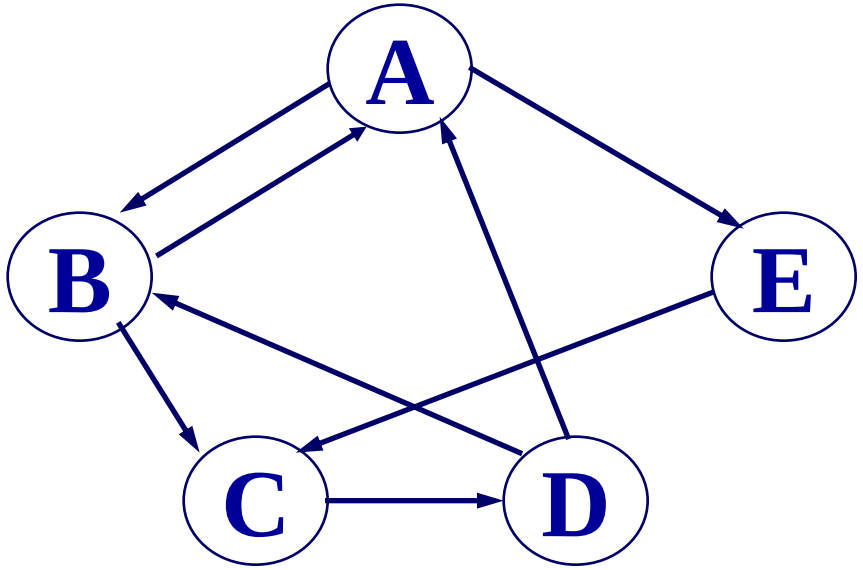
$E = \{(x, y) \mid x, y \in V\}$ 是顶点之间关系的有穷集合，也叫做边(edge)集合。

# 有向图与无向图

- 在有向图中，顶点对 <x, y> 是有序的。
- 在无向图中，顶点对(x, y)是无序的。
  - 若<v, w>∈E 必有<w, v>∈E

# 有向图的例子

$G_1 = (V_1, E_1)$




其中

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

$E_1$={<A,B>, <B,A> ,
<A,E>, <B,C>, <C,D>,
<D,B>, <D,A>, <E,C> }

# 无向图的例子

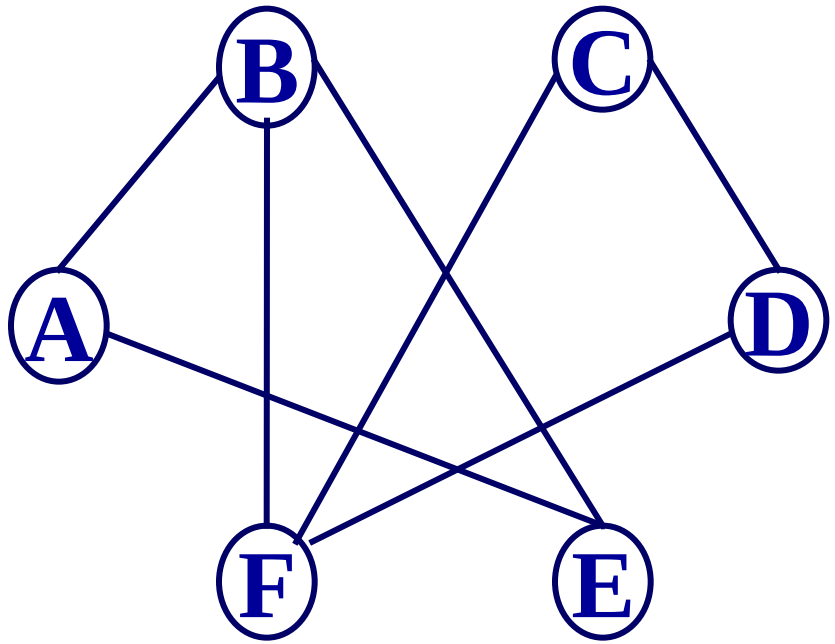
$G_2=(V_2,E_2)$

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

$E_2=\{(A, B), (A, E),$

$(B, E), (C, D), (D, F),$

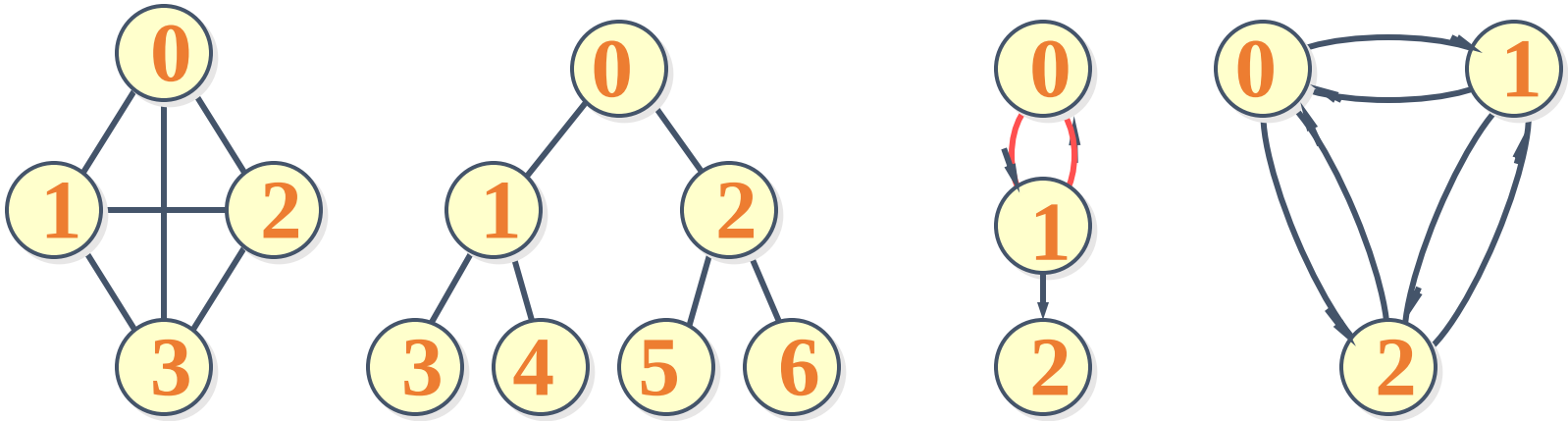
$(B, F), (C, F)\ \}$



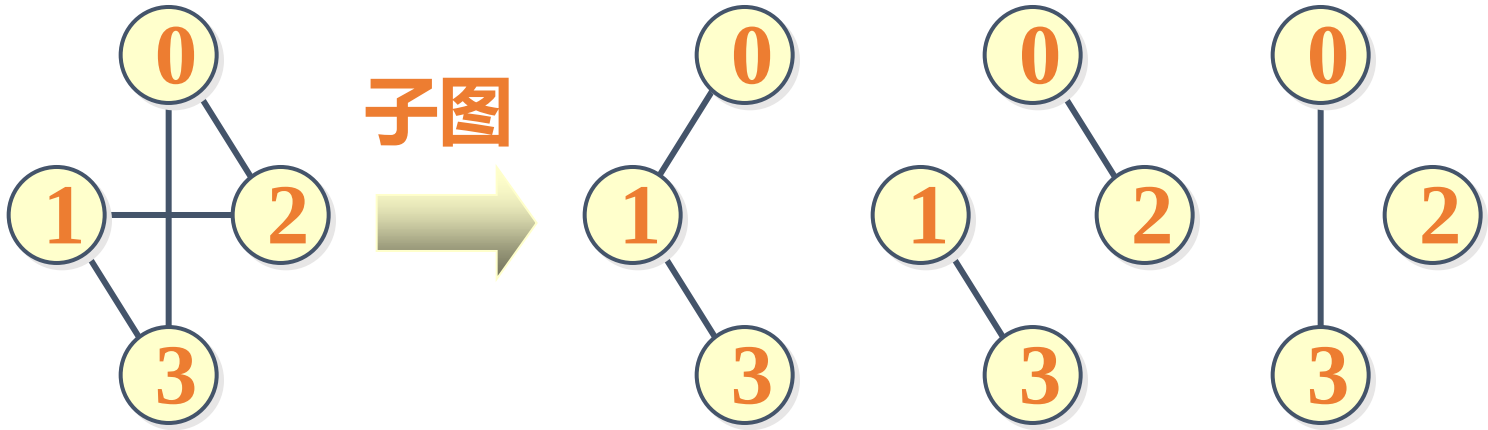
# 完全图

- 若有 *n* 个顶点的无向图有 *n*(*n*-1)/2 条边, 则此图为**完全无向图**。

  有 *n* 个顶点的有向图有*n*(*n*-1) 条边, 则此图为**完全有向图**。

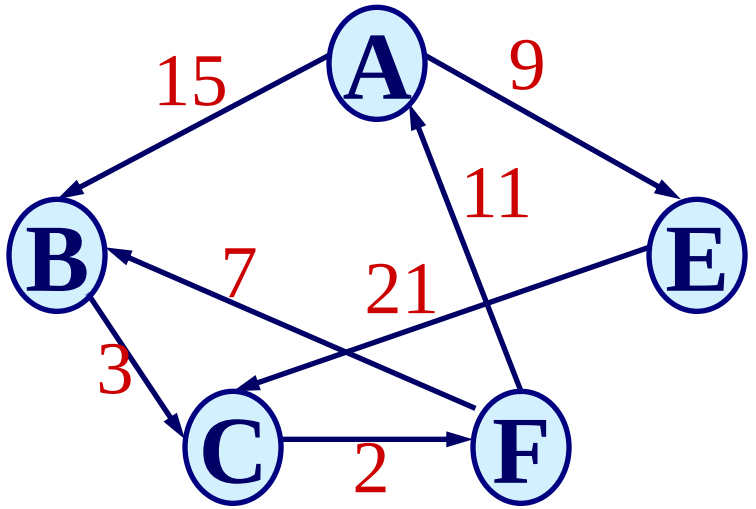


- **邻接顶点** 如果 $(u, v)$ 是 $E(G)$ 中的一条边，则称 $u$ 与 $v$ 互为邻接顶点。
- **子图** 设有两个图 $G = (V, E)$ 和 $G' = (V', E')$。若 $V' \subseteq V$ 且 $E' \subseteq E$，则称 图 $G'$ 是 图G 的子图。



# 网络

- 某些图的边具有与它相关的数，称之为权。这种带权图叫做网络。

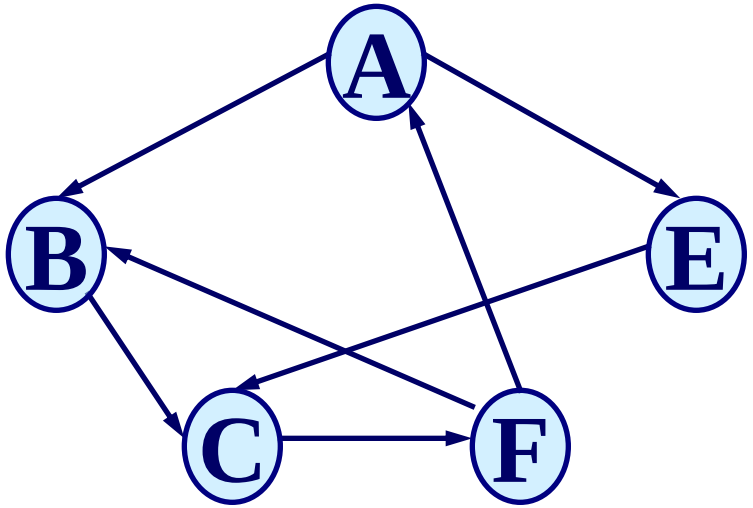


# 顶点的度

- **顶点 *v* 的入度**是以 *v* 为终点的有向边的条数, 记作 ID(*v*)
- **顶点 *v* 的出度**是以 *v* 为始点的有向边的条数, 记作 OD(*v*)
- **顶点的度** 一个顶点*v*的度是与它相关联的边的条数, 记作TD(*v*)。
  - 在有向图中, 顶点的度等于该顶点的入度与出度之和:
  - TD(v)=ID(v)+OD(v)

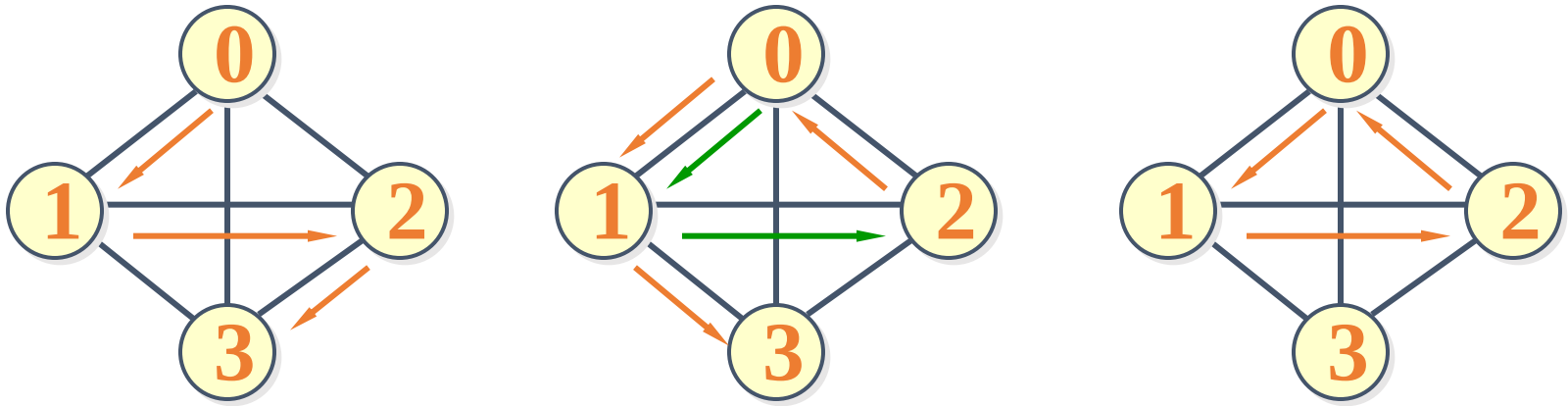
# 路径

- 在图 G = (*V*, *E*) 中, 若从顶点 $v_i$ 出发, 沿一些边经过一些顶点 $v_{p1}, v_{p2}, \dots, v_{pm}$, 到达顶点$v_j$。则称顶点序列 $(v_i\ v_{p_1}\ v_{p_2} \dots v_{p_m} v_j)$ 为从顶点$v_i$到顶点 $v_j$的路径。它经过的边$(v_i, v_{p_1})$、$(v_{p_1}, v_{p_2})$、…、$(v_{p_m}, v_j)$ 应是属于*E*的边。



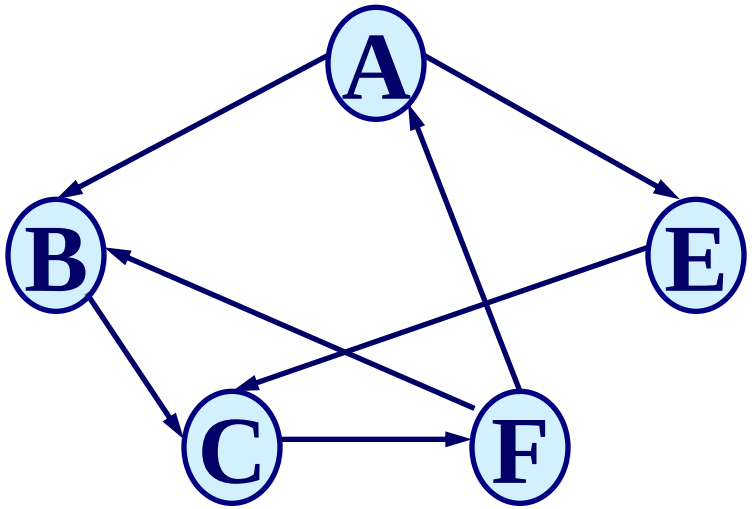

**如:**从A到F长度为 3 的路径$\{A, B, C, F\}$

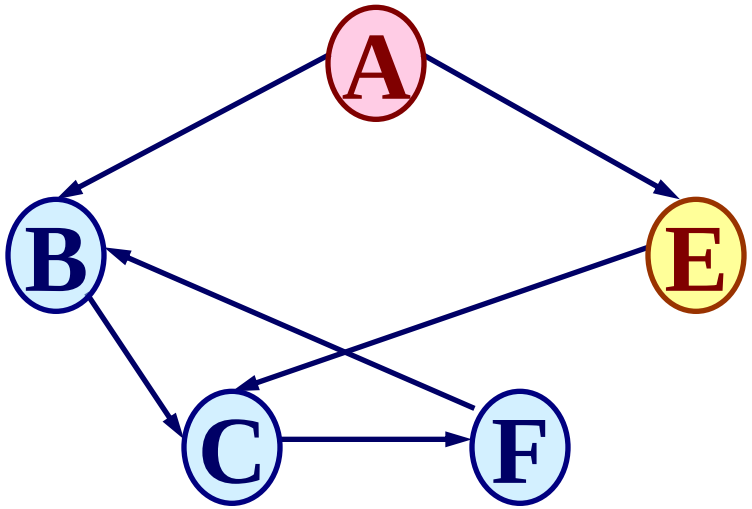
- **路径长度** 非带权图的路径长度是指此路径上边的条数。带权图的路径长度是指路径上各边的权之和。
- **简单路径** 若路径上各顶点 $v_1, v_2, \ldots, vm$均不互相重复，则称这样的路径为简单路径。
- **回路** 若路径上第一个顶点 $v_1$ 与最后一个顶点$v_m$ 重合，则称这样的路径为回路或环。



# 强连通图

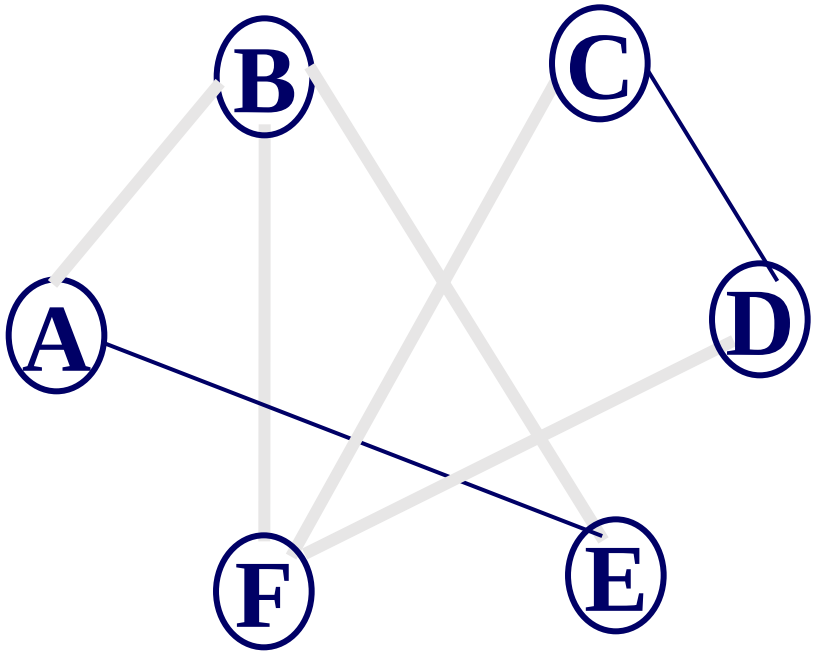
- **强连通图** 在有向图中，若对于每一对顶点$v_i$和$v_j$，都存在一条从$v_i$到$v_j$和从$v_j$到$v_i$的路径，则称此图是强连通图。
- **强连通分量** 非强连通图的极大强连通子图叫做强连通分量。



# 生成树

- **生成树** 一个连通图的生成树是其极小连通子图，在*n*个顶点的情形下，有*n*-1条边。




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

# 图的基本操作

- 图结构的建立和销毁
- 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

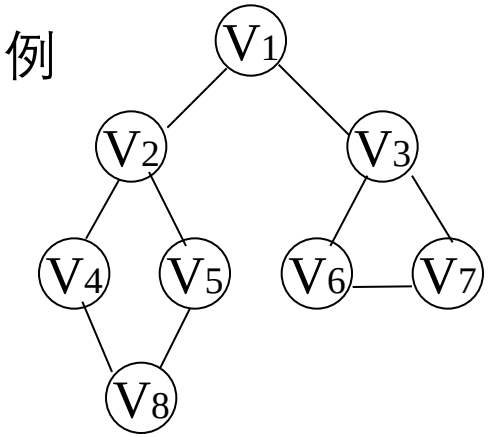
# 图的抽象数据类型

```
class Graph
{
public:
    Graph ( );
    void InsertVertex ( Type & vertex );
    void InsertEdge ( int v1,  int v2,  int weight );
    void RemoveVertex ( int v );
    void RemoveEdge ( int v1,  int v2 );
    int IsEmpty ( );
    Type GetWeight ( int v1,  int v2 );
    int GetFirstNeighbor ( int v );
    int GetNextNeighbor ( int v1,  int v2 );
}
```

# 图的遍历(Graph Traversal )

- 从图中某个顶点出发游历图，访遍图中其余顶点，并且使图中的每个顶点仅被访问一次的过程

例




遍历1：V1⇒ V2 ⇒V4 ⇒ V8 ⇒V5 ⇒V3 ⇒V6 ⇒V7

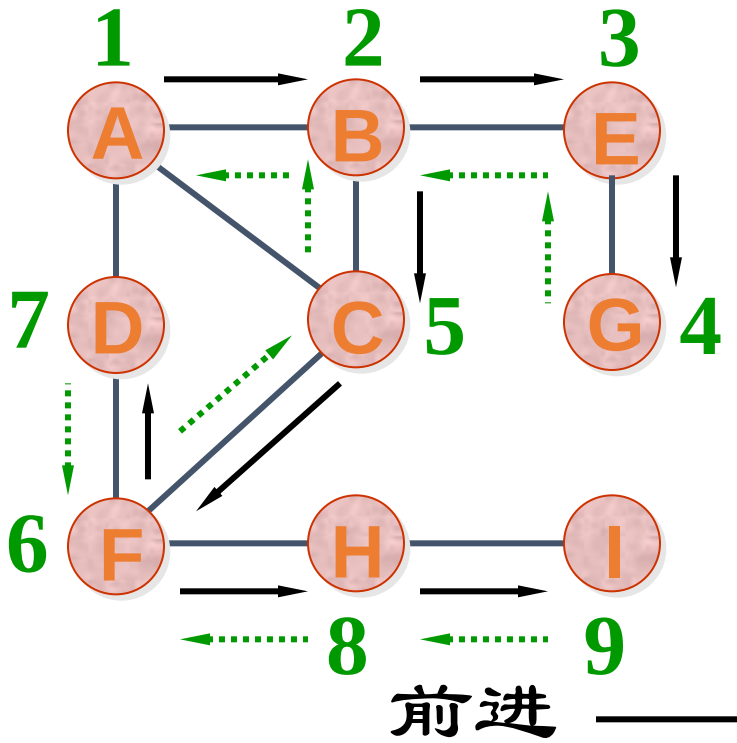
遍历2：V1⇒ V2 ⇒V3 ⇒ V4 ⇒V5 ⇒V6 ⇒V7 ⇒V8

# 图的遍历分类

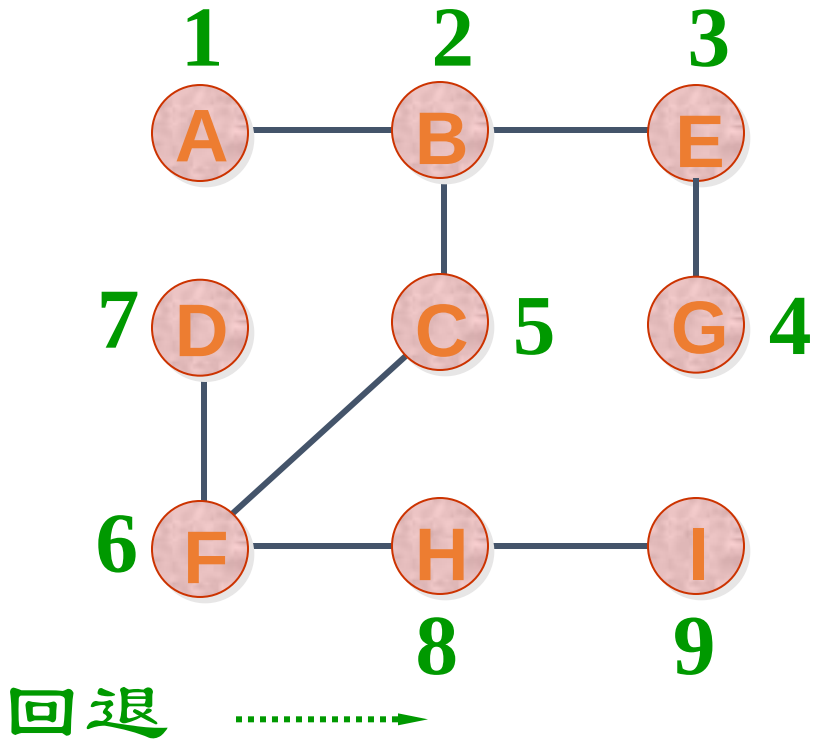
- **深度优先搜索**
  **DFS (Depth First Search)**
- **广度优先搜索**
  **BFS (Breadth First Search)**

# 深度优先搜索DFS

## 深度优先搜索的示例




深度优先搜索过程

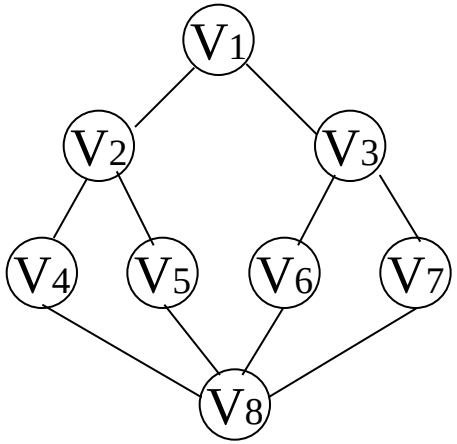



深度优先生成树

前进 ——→　回退 ⋯⋯→

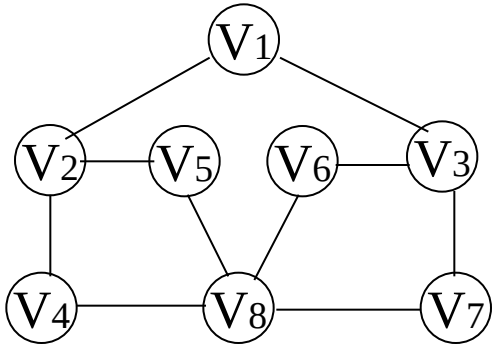
- DFS 在访问图中某一起始顶点 $v$ 后
- 由 $v$ 出发, 访问它的任一邻接顶点 $w_1$; 再从 $w_1$ 出发,访问与 $w_1$ 邻 接但还没有访问过的顶点 $w_2$; 然后再从 $w_2$ 出发, 进行类似的访问, …
- 如此进行下去，直至到达所有的邻接顶点都被访问过的顶点 $u$ 为止。
- 接着，退回一步，退到前一次刚访问过的顶点，看是否还有其它没有被访问的邻接顶点。如果有，则访问此顶点, 之后再从此顶点出发，进行与前述类似的访问; 如果没有, 就再退回一步进行搜索。
- 重复上述过程, 直到连通图中所有顶点都被访问过为止。

# 例子




深度遍历：V1⇒ V2 ⇒V4 ⇒ V8 ⇒V5 ⇒V6 ⇒V3 ⇒V7

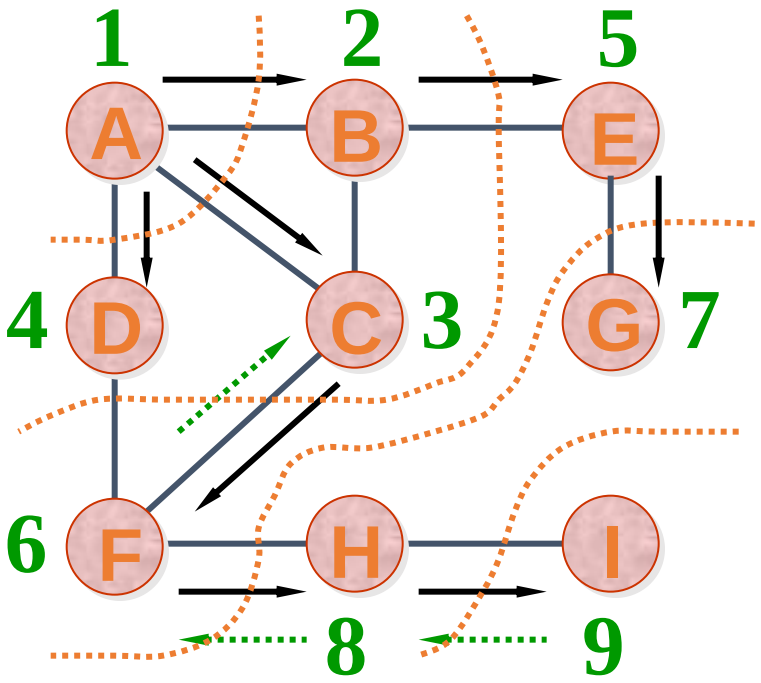
例



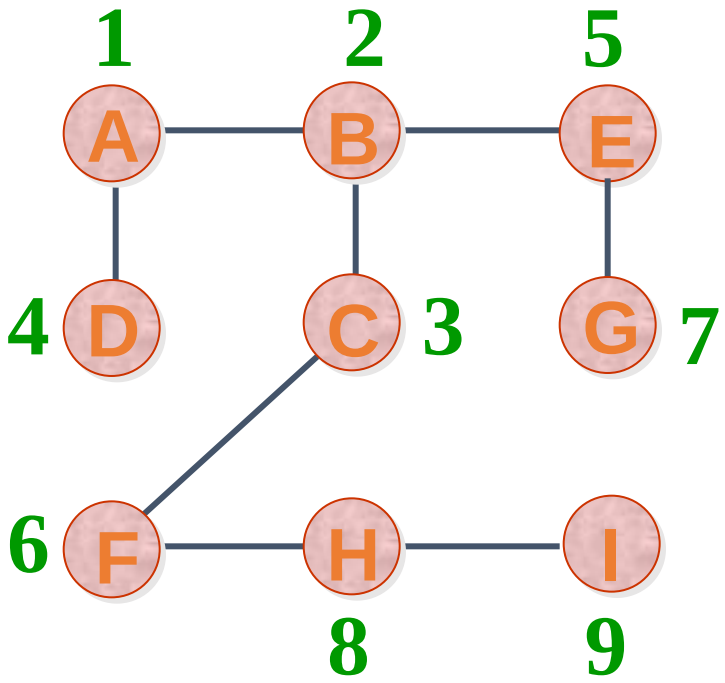

深度遍历：V1⇒ V2 ⇒V4 ⇒ V8 ⇒V5 ⇒V6 ⇒V3 ⇒V7

# 广度优先搜索BFS

- **广度优先搜索的示例**




**广度优先搜索过程**

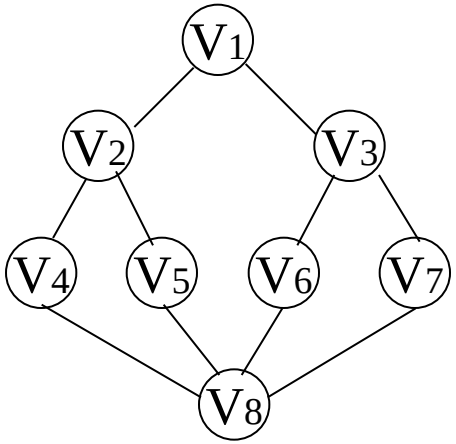



**广度优先生成树**

# 算法描述

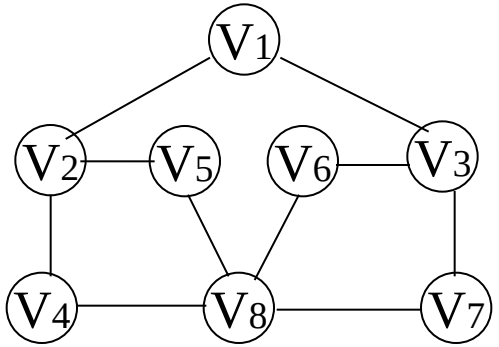
- BFS在访问了起始顶点 $v$ 之后
- 由 $v$ 出发, 依次访问 $v$ 的各个未被访问过的邻接顶点 $w_1, w_2, \ldots, w_t$,
- 然后再顺序访问 $w_1, w_2, \ldots, w_t$ 的所有还未被访问过的邻接顶点。
- 再从这些访问过的顶点出发，再访问它们的所有还未被访问过的邻接顶点，…如此做下去，直到图中所有顶点都被访问到为止。

# 例子




广度遍历：V1⇒ V2 ⇒V3 ⇒ V4 ⇒V5 ⇒V6 ⇒V7 ⇒V8

例



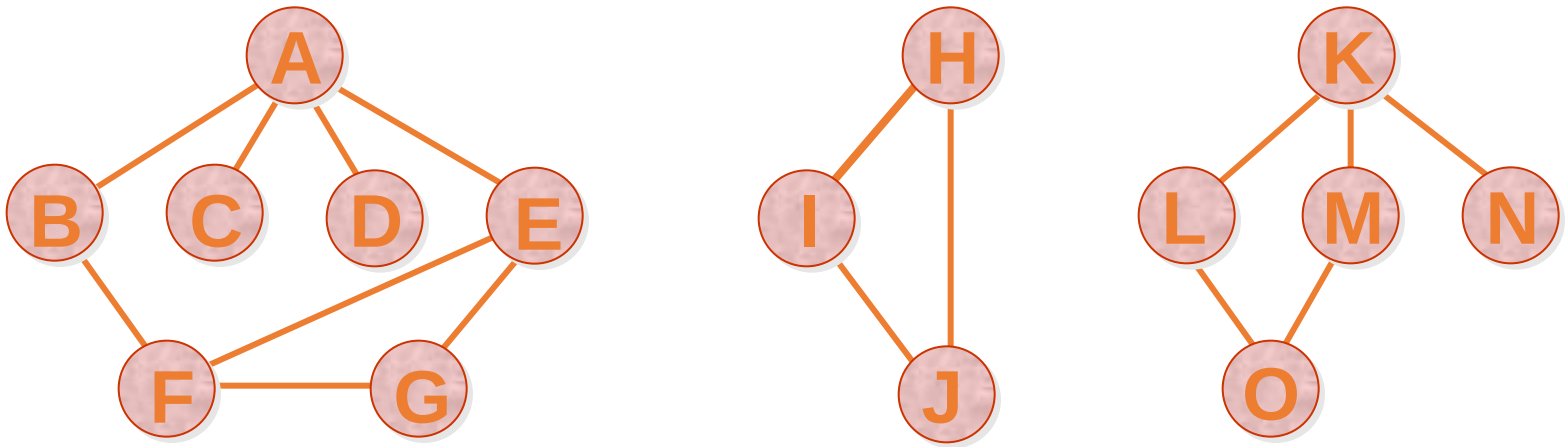

广度遍历：V1⇒ V2 ⇒V3 ⇒ V4 ⇒V5 ⇒V6 ⇒V7 ⇒V8

# 非连通图的连通分量

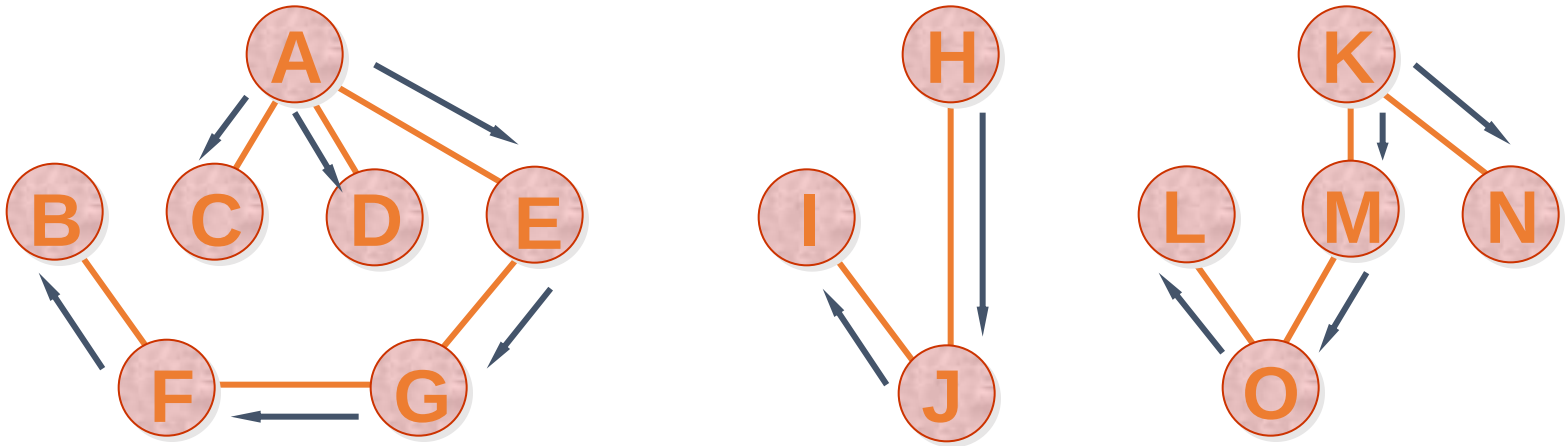
- 当无向图为非连通图时，从图中某一顶点出发，利用深度优先搜索算法或广度优先搜索算法不可能遍历到图中的所有顶点，只能访问到该顶点所在的最大连通子图(连通分量)的所有顶点。

# 连通分量

- 若从无向图的每一个连通分量中的一个顶点出发进行遍历，可求得无向图的所有连通分量。
- 在算法中，需要对图的每一个顶点进行检测：若已被访问过，则该顶点一定是落在图中已求得的连通分量上；若还未被访问，则从该顶点出发遍历图，可求得图的另一个连通分量。
- 对于非连通的无向图，所有连通分量的生成树组成了非连通图的生成森林。




非连通无向图




非连通图的连通分量

# 最小生成树

minimum cost spanning tree

# 最小生成树

- 使用不同的遍历图的方法，可以得到不同的生成树；
- 从不同的顶点出发，也可能得到不同的生成树。

- 按照生成树的定义，$n$ 个顶点的连通网络的生成树有 $n$ 个顶点、$n$-1 条边。

# 构造最小生成树的准则

- 必须使用且仅使用该网络中的$n-1$ 条边来联结网络中的 $n$ 个顶点；
- 不能使用产生回路的边；
- 各边上的权值的总和达到最小

# 最短路径

Shortest path

# 最短路径问题

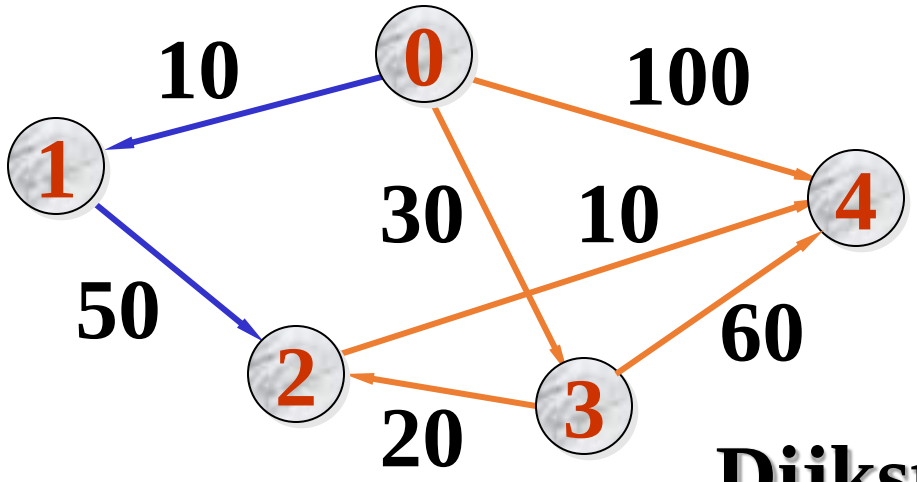
- 如果从图中某一顶点(称为源点)到达另一顶点(称为终点)的路径可能不止一条，如何找到一条路径使得沿此路径上各边上的权值总和达到最小

# 算法

- 边上权值非负情形的单源最短路径问题
  - Dijkstra算法
- 边上权值为任意值的单源最短路径问题
  - Bellman和Ford算法
- 所有顶点之间的最短路径
  - Floyd算法

# 边上权值非负情形的单源最短路径问题

- **问题：** 给定一个带权有向图*D*与源点 *v*，求从 *v* 到*D*中其它顶点的最短路径。限定各边上的权值大于或等于0。
- 为求得这些最短路径, Dijkstra提出按路径长度的递增次序，逐步产生最短路径的算法。
- 首先求出长度最短的一条最短路径，再参照它求出长度次短的一条最短路径，依次类推，直到从顶点v到其它各顶点的最短路径全部求出为止。




## Dijkstra逐步求解的过程

| 源点 | 终点 | 最短路径 | 路径长度 |
|---|---|---|---|
| $v_0$ | $v_1$ | $(v_0,v_1)$ | 10 |
| | $v_2$ | —— $(v_0,v_1,v_2)$ $(v_0,v_3,v_2)$ | ∞,60,50 |
| | $v_3$ | $(v_0,v_3)$ | 30 |
| | $v_4$ | $(v_0,v_4)$ $(v_0,v_3,v_4)$ $(v_0,v_3,v_2,v_4)$ | 100,90,60 |

- 引入辅助数组dist。它的每一个分量dist[i]表示当前找到的从源点 $v_0$到终点 $v_i$ 的最短路径的长度。初始状态：
  - 若从源点$v_0$到顶点 $v_i$ 有边，则dist[i]为该边上的权值；
  - 若从源点$v_0$到顶点 $v_i$ 无边，则dist[i]为∞ 。
- 假设 $S$ 是已求得的最短路径的终点的集合，则可证明：下一条最短路径必然是从$v_0$ 出发，中间只经过 $S$ 中的顶点便可到达的那些顶点$v_x$ ($v_x \in V\text{-}S$ )的路径中的一条。
- 每次求得一条最短路径后，其终点$v_k$ 加入集合$S$，然后对所有的$v_i \in V\text{-}S$，修改其 dist[i]值。

# Dijkstra算法可描述如下：

① 初始化： $S \leftarrow \{ v_0 \}$;

dist[j] ← Edge[0][j]， j = 1, 2, …, $n$-1; // $n$为图中顶点个数

② 求出最短路径的长度：

dist[$k$] ← min { dist[$i$] }, $i \in V\text{-}S$ ;

$S \leftarrow S \cup \{ k \}$;

③ 修改：

dist[$i$] ← min{ dist[$i$], dist[$k$] + Edge[$k$][$i$] },

对于每一个 $i \in V\text{-}S$ ;

④ 判断：若 $S = V$, 则算法结束，否则转 ②。

# Sensor Network Localization
# 传感器网络定位

# 问题

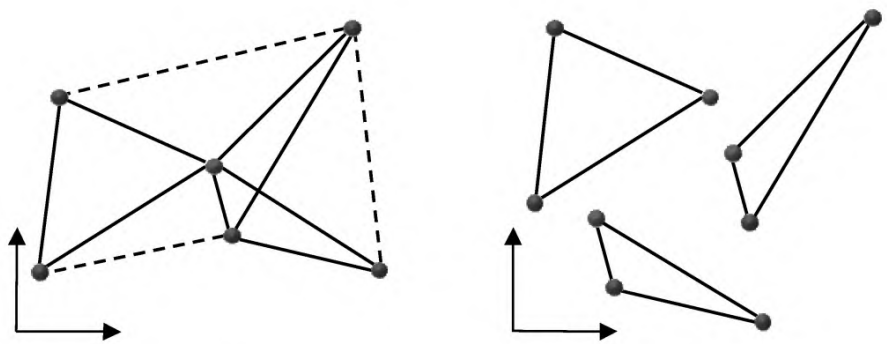
- 背景:
  - 无线传感器网络（军事、工业、环境、医疗等）
  - 分子结构预测
  - 互联网拓扑关系
  - 地图重建
  - 数据可视化
- 问题:
  - 每个传感器能感知到其周边一定范围内的其他传感器的距离，求整体的传感器分布结构
  - 从局部距离出发，推断整体结构

# 数学模型

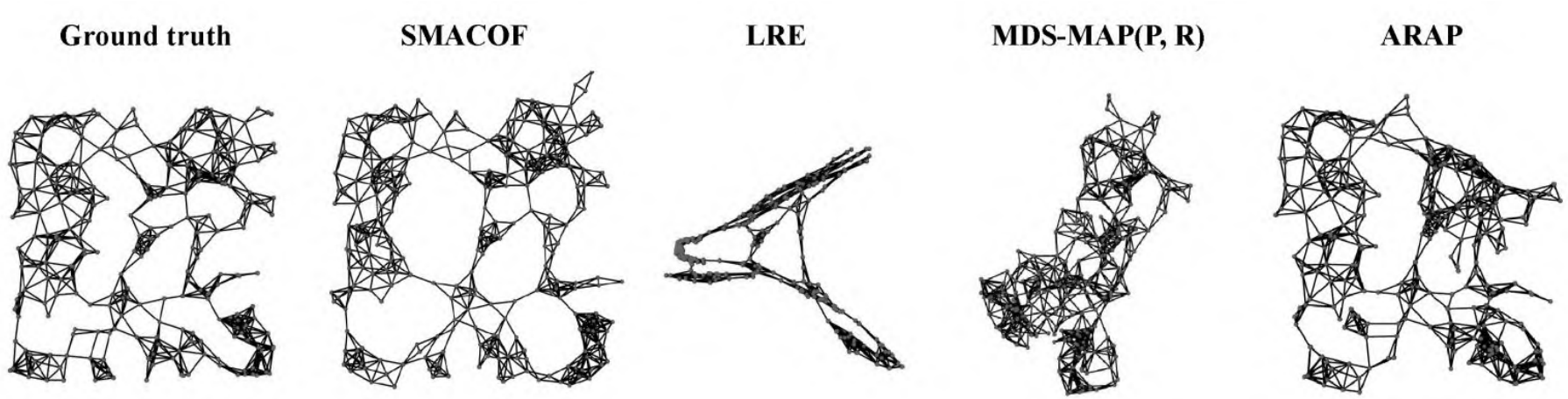
- 优化模型

$$Stress(p_1, \ldots, p_n) = \sum_{(i,j)\in E} (\|p_i - p_j\| - d_{ij})^2$$

- 几何模型



# 不同方法的比较



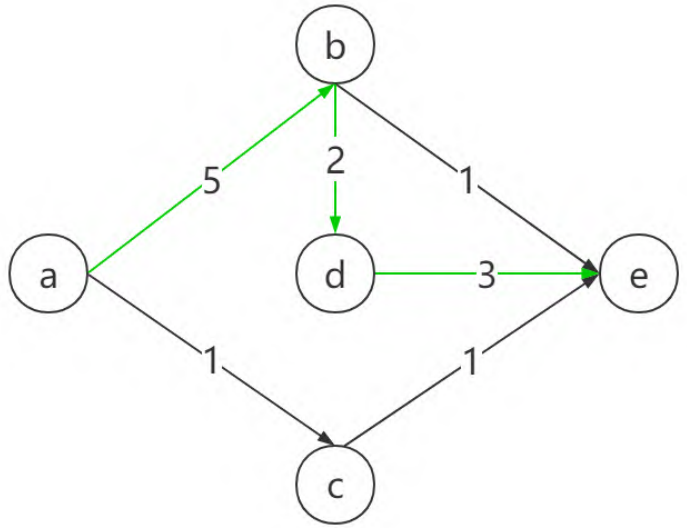
# 最大流最小割

# 最大流最小割定理

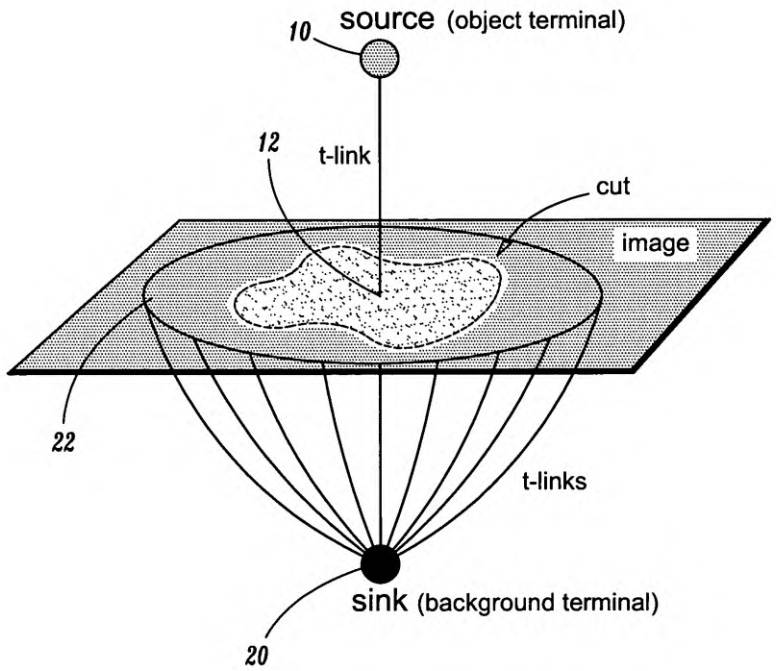
- 在一个网络流中，能够从源点到达汇点的最大流量等于如果从网络中移除就能够导致网络流中断的边的集合的最小容量和。即在任何网络中，最大流的值等于最小割的容量。
- 背景:
  - 现实生活中，人们经常见到一些网络，如铁路网、公路网、通信网、运输网等等。这些网络有一个共同的特点，就是在网络中都有物资、人或信息等某种量从一个地方流向另一个地方，如何安排这些量的流动以便取得最大效益是一个很有意义的实际问题。

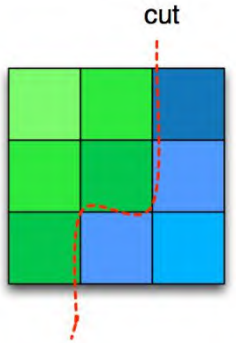
# 最大流和最小割

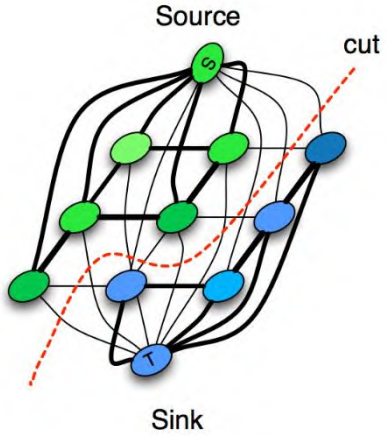
- 最大流
  - 给定指定的一个有向图，其中有两个特殊的点源S(Sources)和汇T(Sinks)，每条边有指定的容量(Capacity)，求满足条件的从S到T的最大流(MaxFlow)。
- 最小割
  - 割是网络中定点的一个划分，它把网络中的所有顶点划分成两个顶点集合S和T，其中源点s∈S,汇点t∈T。记为CUT(S,T)，满足条件的从S到T的最小割（Min cut）。

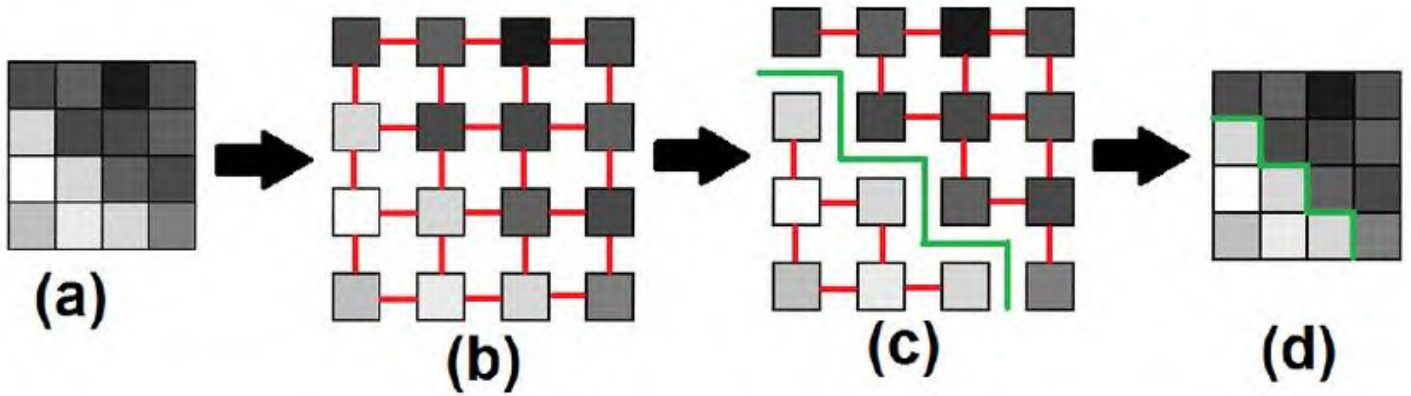


# Graphcut


10
source (object terminal)
12
t-link
cut
image
22
t-links
sink (background terminal)
20


cut


Source
cut
S
T
Sink


(a)
(b)
(c)
(d)

# Image Segmentation

Lazy Snapping

GrabCut

# 更多例子

## 例3：（CUMCM2007B ）乘公交看奥运。

我国人民翘首企盼的第29届奥运会明年8月将在北京举行，届时有大量观众到现场观看奥运比赛，其中大部分人将会乘坐公共交通工具（简称公交，包括公汽、地铁等）出行。这些年来，城市的公交系统有了很大发展，北京市的公交线路已达800条以上，使得公众的出行更加通畅、便利，但同时也面临多条线路的选择问题。针对市场需求，某公司准备研制开发一个解决公交线路选择问题的自主查询计算机系统。

为了设计这样一个系统，其核心是线路选择的模型与算法，应该从实际情况出发考虑，满足查询者的各种不同需求。请你们解决如下问题:

1、仅考虑公汽线路，给出任意两公汽站点之间线路选择问题的一般数学模型与算法。并根据附录数据，利用你们的模型与算法，求出以下6对起始站→终到站之间的最佳路线（要有清晰的评价说明）。

(1) S3359→S1828　(2) S1557→S0481　(3) S0971→S0485

(4) S0008→S0073　(5) S0148→S0485　(6) S0087→S3676

2、同时考虑公汽与地铁线路，解决以上问题。

3、假设又知道所有站点之间的步行时间，请你给出任意两站点之间线路选择问题的数学模型。

【附录1】基本参数设定

相邻公汽站平均行驶时间(包括停站时间)： 3分钟

相邻地铁站平均行驶时间(包括停站时间)： 2.5分钟

公汽换乘公汽平均耗时： 5分钟(其中步行时间2分钟)

地铁换乘地铁平均耗时： 4分钟(其中步行时间2分钟)

地铁换乘公汽平均耗时： 7分钟(其中步行时间4分钟)

公汽换乘地铁平均耗时： 6分钟(其中步行时间4分钟)

公汽票价：分为单一票价与分段计价两种，标记于线路后；其中分段计价的票价为：0～20站1元；21～40站2元；40站以上3元

地铁票价：3元（无论地铁线路间是否换乘）

注：以上参数均为简化问题而作的假设，未必与实际数据完全吻合。

【附录2】公交线路及相关信息 （见数据文件B2007data.rar）

【模型建立】

- 以公汽和地铁站点为顶点构造有向图$G = (V, E)$，其中$\overrightarrow{uv} \in E \Leftrightarrow$ 可乘某一条线路从$u$到$v$或者从$u$换乘到$v$。
- $|V| = 3996$，$|E| = 547050$，两个顶点之间可能有多条边。
- 对于每条边$e \in E$，定义$L(e) = (l_1, l_2, l_3)$，分别代表总时间、步行时间、票价。
- 数学问题：求任意两个顶点之间以 $\sum_{e \in P} L(e)$ 为优化目标的最优路径$P$。

## 例4：（CUMCM2017D）巡检线路的排班。

某化工厂有26个点需要进行巡检以保证正常生产，各个点的巡检周期、巡检耗时、两点之间的连通关系及行走所需时间在附件中给出。

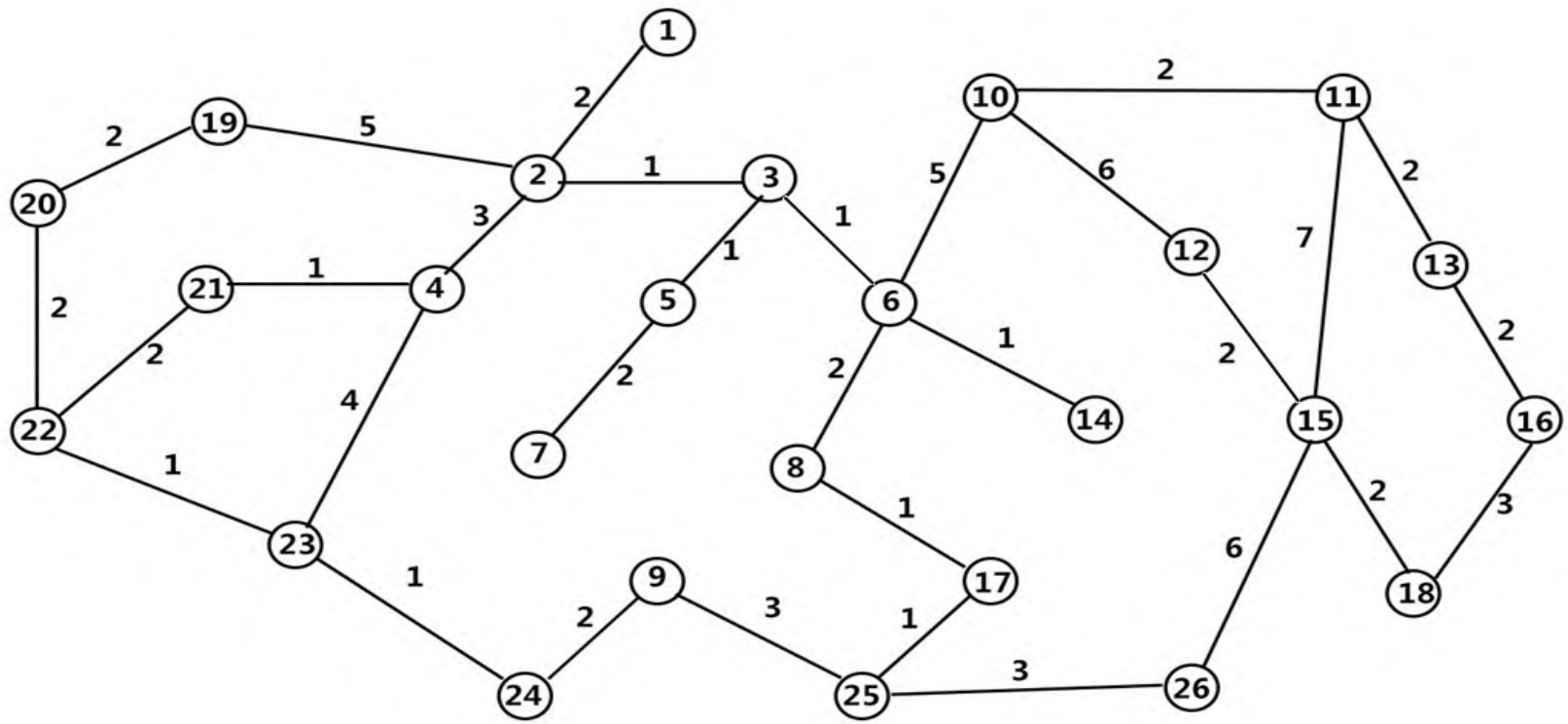
每个点每次巡检需要一名工人，巡检工人的巡检起始地点在巡检调度中心（XJ0022），工人可以按固定时间上班，也可以错时上班，在调度中心得到巡检任务后开始巡检。现需要建立模型来安排巡检人数和巡检路线，使得所有点都能按要求完成巡检，并且耗费的人力资源尽可能少，同时还应考虑每名工人在一时间段内（如一周或一月等）的工作量尽量平衡。

问题1. 如果采用固定上班时间，不考虑巡检人员的休息时间，采用每天三班倒，每班工作8小时左右，每班需要多少人，巡检线路如何安排，并给出巡检人员的巡检线路和巡检的时间表。

问题2. 如果巡检人员每巡检2小时左右需要休息一次，休息时间大约是5到10分钟，在中午12时和下午6时左右需要进餐一次，每次进餐时间为30分钟，仍采用每天三班倒，每班需要多少人，巡检线路如何安排，并给出巡检人员的巡检线路和巡检的时间表。

问题3. 如果采用错时上班，重新讨论问题1和问题2，试分析错时上班是否更节省人力。

| 位号 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 周期 | 35 | 50 | 35 | 35 | 720 | 35 | 80 | 35 | 35 | 120 | 35 | 35 |
| 耗时 | 3 | 2 | 3 | 2 | 2 | 3 | 2 | 3 | 4 | 2 | 3 | 2 |
| 位号 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| 周期 | 80 | 35 | 35 | 35 | 480 | 35 | 35 | 35 | 80 | 35 | 35 | 35 |
| 耗时 | 5 | 3 | 2 | 3 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 2 |



【模型建立】

- 建立附件所示的无向图$G=(V,E)$，$|V|=26$。
- 固定上班时间与错时上班并无区别。关键是上一班的工人何时下班。
- 设所有人的巡检路线都是$x_0 \to x_1 \to \cdots \to x_{26}$，其中$x_0 = x_{26} = 22$。
- 巡检时间$T = \sum_{i=1}^{26}$巡检耗时$(x_i)$ + 行走时间$(x_{i-1}, x_i)$。
- 问题化为求经过所有顶点的回路，使得$T$最小。
- 工人只需间隔35分钟从调度中心出发即可满足题设。
- 若考虑休息时间$r$，则间隔时间改为$35-r$。
- 不可能所有工人同时集中进餐30分钟。
- 经计算，存在巡检时间139分钟的路线（非最优），每班4人即可。

## 例5：（CUMCM2013B）碎纸片的拼接复原。

破碎文件的拼接在司法物证复原、历史文献修复以及军事情报获取等领域都有着重要的应用。传统上，拼接复原工作需由人工完成，准确率较高，但效率很低。特别是当碎片数量巨大，人工拼接很难在短时间内完成任务。随着计算机技术的发展，人们试图开发碎纸片的自动拼接技术，以提高拼接复原效率。请讨论以下问题:

1. 对于给定的来自同一页印刷文字文件的碎纸机破碎纸片（仅纵切），建立碎纸片拼接复原模型和算法，并针对附件1、附件2给出的中、英文各一页文件的碎片数据进行拼接复原。如果复原过程需要人工干预，请写出干预方式及干预的时间节点。复原结果以图片形式及表格形式表达（见【结果表达格式说明】）。

2. 对于碎纸机既纵切又横切的情形，请设计碎纸片拼接复原模型和算法，并针对附件3、附件4给出的中、英文各一页文件的碎片数据进行拼接复原。如果复原过程需要人工干预，请写出干预方式及干预的时间节点。复原结果表达要求同上。

3. 上述所给碎片数据均为单面打印文件，从现实情形出发，还可能有双面打印文件的碎纸片拼接复原问题需要解决。附件5给出的是一页英文印刷文字双面打印文件的碎片数据。请尝试设计相应的碎纸片拼接复原模型与算法，并就附件5的碎片数据给出拼接复原结果，结果表达要求同上。

【数据文件说明】

(1) 每一附件为同一页纸的碎片数据。

(2) 附件1、附件2为纵切碎片数据，每页纸被切为19条碎片。

(3) 附件3、附件4为纵横切碎片数据，每页纸被切为11×19个碎片。

(4) 附件5为纵横切碎片数据，每页纸被切为11×19个碎片，每个碎片有正反两面。该附件中每一碎片对应两个文件，共有2×11×19个文件，例如，第一个碎片的两面分别对应文件000a、000b。

【结果表达格式说明】

复原图片放入附录中，表格表达格式如下：

(1) 附件1、附件2的结果：将碎片序号按复原后顺序填入1×19的表格；

(2) 附件3、附件4的结果：将碎片序号按复原后顺序填入11×19的表格；

(3) 附件5的结果：将碎片序号按复原后顺序填入两个11×19的表格；

(4) 不能确定复原位置的碎片，可不填入上述表格，单独列表。

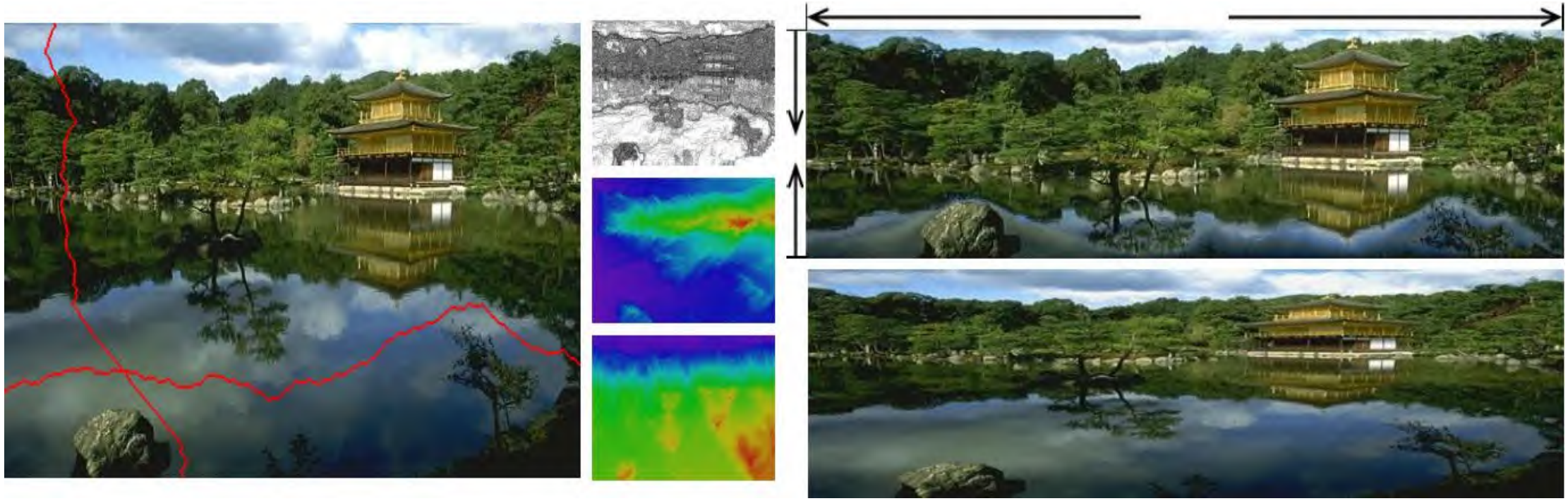
【模型建立】

- 需要解决的实际问题：求碎片的排列（对于双面打印的碎片可能还需要翻转），使得相邻的碎片可以拼接起来。
- 对于仅纵切情形。以碎片为顶点构造有向图$G=(V,E)$，其中$\overrightarrow{uv}\in E$ ⇔碎片$u$的右边界与碎片$v$的左边界可以拼接。
- 对于任意两块碎片$x,y$，可用一个函数$f(x,y)$描述它们的拼接吻合程度。问题转化为求图$G$中的一条经过所有顶点的有向Path，使得$\sum_{e\in P} f(e)$最大。
- 对于既纵切又横切的情形。定义函数$g(x,y)$判断两块碎片$x,y$是否属于相同行，并描述它们的拼接吻合程度。据此把$m\times n$个碎片平均分成$m$部分。问题转化为纵切和横切情形。
- 对于双面情形。定义两块碎片$x,y$的拼接吻合程度函数

$$h(x,y)=\max(g(x,y),g(x,\tilde{y}),g(\tilde{x},y),g(\tilde{x},\tilde{y}))$$

# Seam Carving

- **思想：**通过在图像中增加水平和竖直方向连续的缝隙进行扩大或缩小图像
- seam：图像中连通的低能量像素通路，并且每行或者每列只包含一个像素



中国科学技术大学

University of Science and Technology of China

# 谢　谢！