# 定义

顶点、弧（弧头、弧尾）、出度、入度、边、邻接点

# 分类

## 无向图

### 连通图

### 完全图

### 生成树

## 有向图

# 存储

## 邻接矩阵

## 邻接表

## 十字链表

## 邻接多重表

## 边集数组

# 遍历

深度优先遍历简称DFS（Depth First Search），广度优先遍历简称BFS（Breadth First Search），它们是遍历图当中所有顶点的两种方式。

## 深度优先

### 概述

先深入探索，走到头再回退寻找其他出路的遍历方式，就叫做深度优先遍历（DFS）。

注：二叉树的前序、中序、后序遍历，本质上也可以认为深度优先遍历。

### 算法

实现深度优先遍历的关键在于“回溯”，实现广度优先遍历的关键在于“重放”。

回溯就是自后向前，追溯曾经走过的路径。

要想实现回溯，可以利用栈的先入后出特性，也可以采用递归的方式（因为递归本身就是基于方法调用栈来实现）。

## 广度优先

### 概述

一层一层由内而外的遍历方式，就叫做广度优先遍历（BFS）。

注：二叉树的层次遍历本质上也可以认为是广度优先遍历。

### 算法

把遍历过的顶点按照之前的遍历顺序重新回顾，就叫做重放。同样的，要实现重放也需要额外的存储空间，可以利用队列的先入先出特性来实现。

/\*\*

\* 图的顶点

\*/

private static class Vertex

{

int data;

Vertex(int data) {

this.data = data;

}

}

# 应用

## 最小生成树

## 最短路径

## 拓扑排序

## 关键路径