C++ 算法：图论（一）- 图的基本概念与模型 讲义

一、 引言

1.1 问题引入：从校园地图到数据结构

我们已经学习了线性结构（如数组、链表）和树形结构（如二叉树）。前者模拟了“一对一”的关系，后者模拟了“一对多”的层次关系。但现实世界充满了更为复杂的联系。

例如校园地图：教学楼、食堂、图书馆、体育馆等建筑散布其间，复杂的道路网将它们连接起来。现在，我们面临一些实际问题：

- 如何找到从大门口到图书馆的**最短路径**？
- 如何规划一条路线，可以**不重复地走完**所有关键道路？
- 如果某个路口正在施工，从A到B**是否还有其他路径**可以到达？

这些问题，都无法用简单的线性或树形结构来描述。我们需要一种新的数据结构，它能专门模拟这种“多对多”的复杂网络关系。这个结构，就是我们今天要学习的——**图（Graph）**。

1.2 价值阐述：为什么图论很重要?

图，是描述“多对多”关系的最强模型。它的应用无处不在：

- **社交网络**：每个人是一个点，好友关系是一条线，构成一张巨大的社交图。
- **交通网络**：城市是点，公路、铁路、航线是线，构成了交通图。导航软件的核心就是图算法。
- **计算机网络**：路由器、服务器是点，网线是线，构成了计算机网络拓扑图。

二、 核心思想：从具象到抽象

2.1 思路铺垫

对于校园地图的例子。

- 地图上的每一个关键**地点**（大门、教学楼、食堂...），我们把它抽象成一个**点**。
- 连接两个地点之间的每一条**道路**，我们把它抽象成一条**线**。

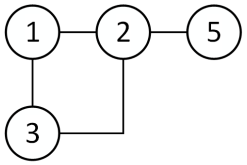
通过这种方式，一张现实地图，就被简化成了一个由“点”和“线”组成的数学图形。这就是图的本质。

2.2 图的数学定义

在数学和计算机科学中，我们对图有更精确的定义：
一个图 `G` 是一个二元组 `(V, E)`，记为 `G = (V, E)`。

- `V` **(Vertex Set)**：一个非空的**顶点**集合。每个顶点代表一个实体或对象。
- `E` **(Edge Set)**：一个**边**的集合。每条边连接 `V` 中的两个顶点，代表它们之间存在某种连接关系。

示例：



对于上图 `G`：

- 顶点集合 V = {1, 2, 3, 5}
- 边集合 E = {(1, 2), (1, 3), (2, 3), (2, 5)}

2.3 图的核心组成元素

1. **顶点 (Vertex)**
    - 也称为**节点**。在图中，它就是那些独立的“点”，用于代表现实世界中的事物。
    - **例如**：校园地图中的“建筑物”，社交网络中的“用户”，交通网络中的“城市”。
2. **边 (Edge)**
    - 连接两个顶点的“线”，用于表示两个顶点（事物）之间的特定关系。
    - **例如**：地图中的“道路”，社交网络中的“好友关系”，交通网络中的“航线”。

三、 图的分类与属性

现实世界的关系是多种多样的，因此图也有不同的类型和属性来描述它们。

3.1 无向图

- **定义**：图中的每条边都是**双向**的。如果从顶点u到顶点v有一条边，那么也必然可以从v回到u。边的表示用圆括号 `(u, v)`，且 `(u, v)` 与 `(v, u)` 代表同一条边。
- **现实类比**：双向通行的马路、互为好友的社交关系。
- **相关术语：度**
    - 一个顶点的**度**，是指与该顶点**直接相连**的边的数量。
    - **示例**：在 讲义 **2.2 节** 的图中，顶点2的度为3，因为它连接了1、3、5三个顶点。

3.2 有向图

- **定义**：图中的每条边都是**单向**的，具有明确的方向性。边的表示用括号来表示 `<u, v>`，它表示一条从u出发，指向v的边，这与 `<v, u>` 是两条完全不同的边。
- **类比**：单行道、社交网络中的“关注”关系（A关注B，不代表B一定关注A）。
- **相关术语：入度 与 出度**
    - 一个顶点的**入度**：指**指向**该顶点的边的数量。
    - 一个顶点的**出度**：指由该顶点**出发**指向其他顶点的边的数量。

3.3 带权图

- **定义**：图中的每条边都被赋予一个数值，这个数值称为**权重 或**成本。这个权重可以代表各种现实意义。
- **边权**：边的权重。
    - **例如**：地图中道路的**长度**或通过所需**时间**；网络中传输数据的**带宽**或**延迟**；交通网络中的**票价**或**距离**。
- **点权**：类似地，顶点也可以被赋予权重，代表该点自身的某种属性值（例如，一个城市的人口，一个景点的好评度）。

3.4 简单图

在大多数算法问题中，我们处理的图都是**简单图**，它满足两个条件：

1. 不包含**自环**：即一条连接顶点和其自身的边。
2. 不包含**重边**：即连接同一对顶点的多条边。

我们后续的讨论，如无特殊说明，都将基于简单图。

四、 图的拓扑属性与概念

除了基本的分类，图还具有一些描述其整体结构与连接特性的概念。

4.1 连通性

- **连通**：若图中从顶点u到顶点v存在一条通路（可以经过若干中间顶点），则称 u 和 v 是**连通**的。
- **连通图**：如果一个**无向图**中，**任意两个顶点**都是连通的，那么这个图被称为**连通图。**反之，如果图中存在至少一个“孤岛”，则为**非连通图。**
- **强连通图**：这个概念专属于**有向图。**如果一个有向图中，对**任意两个顶点**u和v，都**同时存在**从u到v的路径 **和** 从v到u的路径，则称该图为**强连通图。**

4.2 完全图

- **定义**：图中**任意两个不同的顶点之间都存在一条边**的图，称为完全图。完全图是边数达到最大值的最“稠密”的图。
- **无向完全图**：一个有n个顶点的无向完全图，其边数为 `e = n * (n - 1) / 2`。
- **有向完全图**：一个有n个顶点的有向完全图，任意两点间都存在方向相反的两条边，其边数为 `e = n * (n - 1)`。

4.3 路径

- **路径**：在图中，从一个起点到一个终点所经过的**顶点序列**，就是一条路径。
- **最短路径**：在一个带权图中，连接两个顶点的所有路径中，**路径上各边权重之和最小**的那一条路径，被称为最短路径。这是图论算法中一个非常核心、经典的问题。

五、 总结

- **图的本质**：图是用于描述事物间“**多对多**”关系的数学模型，由**顶点(Vertex)**和**边(Edge)**构成，即 `G = (V, E)`。
- **核心分类**：
  - **无向图**：边是双向的，关系是相互的。顶点的核心属性是**度。**
  - **有向图**：边是单向的，关系有明确方向。顶点的核心属性是**入度**和**出度。**
- **重要属性**：
  - **带权图**：边上附有数值（**权重**），代表成本、距离、时间等。
- **拓扑概念**：
  - **连通性**：描述了图中顶点间路径的可达性，分为**连通图**、**强连通图**等。
  - **完全图**：一种边最稠密的特殊图，任意两点间都直接相连。
  - **路径**：是图的遍历基础，而**最短路径**是图算法的核心应用之一(之后的课程会详细讲解)。