（又称｜简称｜记作｜简记）

（！）不重不漏

（？）不重

|  |
| --- |
| （和的）笛卡儿积（||）：设、为集合，用中元素为第一元素，中元素为第二元素，构成有序对。所有这样的有序对组成的集合。 |
| （与的）无序积（||）：设、为两集合，。  （||） |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 图：有序三元组。 | | | | | | | | | | |
| ┣ | （图的）顶点集（|||） | | | | | | | | | |
| ┃ | ┣ | （图的）顶点数（阶、（有个顶点的图）阶图||）：。 | | | | | | | | |
| ┃ | ┗ | （图的）顶点（节点、结点||）：的元素。 | | | | | | | | |
| ┃ |  | ┣ | （顶点的）度数（度、次数||）：顶点作为边的端点的次数之和。在有向图中，。 | | | | | | | |
| ┃ |  | ┃ | ┣ | （图的）最小度（||）：。 | | | | | | |
| ┃ |  | ┃ | ┣ | （图的）最大度（||）： | | | | | | |
| ┃ |  | ┃ | ┗ | （图的）度数序列：设，。 | | | | | | |
| ┃ |  | ┣ | 有向图（顶点的）入度（有条边以该顶点为终点||）：顶点作为边的终点的次数之和。（进入顶点的边的数目。）（从顶点引入的边的数目。） | | | | | | | |
| ┃ |  | ┃ | ┣ | （图的）最小入度（||）：。 | | | | | | |
| ┃ |  | ┃ | ┣ | （图的）最大入度（||）： | | | | | | |
| ┃ |  | ┃ | ┗ | （图的）入度序列：。 | | | | | | |
| ┃ |  | ┣ | 有向图（顶点的）出度（有条边以该顶点为起点||）：顶点作为边的始点的次数之和。（离开顶点的边的数目。）（从顶点引出的边的数目。） | | | | | | | |
| ┃ |  | ┃ | ┣ | （图的）最小出度（||）：。 | | | | | | |
| ┃ |  | ┃ | ┣ | （图的）最大出度（||）： | | | | | | |
| ┃ |  | ┃ | ┗ | （图的）出度序列：。 | | | | | | |
| ┃ |  | ┗ | 分类 | | | | | | | |
| ┃ |  |  | ┗ | 按度（？） | | | | | | |
| ┃ |  |  |  | ┗ | 孤立点（顶点是孤立的）：，。 | | | | | |
| ┃ |  |  |  | ┗ | 悬挂顶点：，。 | | | | | |
| ┣ | （图的）边集（|||） | | | | | | | | | |
| ┃ | ┣ | （图的）边数（||）：。 | | | | | | | | |
| ┃ | ┗ | （图的）边（||、无向图 、无向图 、）：的元素。 | | | | | | | | |
| ┃ |  | ┣ | （边的）端点 | | | | | | | |
| ┃ |  | ┣ | 赋权图（边的）权：若将图的每一条边对应一个实数，。 | | | | | | | |
| ┃ |  | ┗ | 分类 | | | | | | | |
| ┃ |  |  | ┗ | 按可能的端点的数量（？） | | | | | | |
| ┃ |  |  |  | ┗ | 超边：连接任意顶点子集的边。 | | | | | |
| ┃ |  |  |  |  | ┗ | 边：连接2个顶点的边。 | | | | |
| ┃ |  |  |  |  |  | ┣ | 按有无方向（！） | | | |
| ┃ |  |  |  |  |  | ┃ | ┣ | 无向图无向边（|边）：，。（与中的无序偶对应的边。） | | |
| ┃ |  |  |  |  |  | ┃ | ┗ | 有向图有向边（弧|边）：与中的有序偶对应的边。 | | |
| ┃ |  |  |  |  |  | ┣ | 按端点关系（？） | | | |
| ┃ |  |  |  |  |  | ┃ | ┗ | 自环（环）：2个端点为同一顶点的边。 | | |
| ┃ |  |  |  |  |  | ┗ | 按端点的度（？） | | | |
| ┃ |  |  |  |  |  |  | ┗ | 悬挂边：悬挂顶点关联的边。 | | |
| ┣ | （图的）关联函数：。 | | | | | | | | | |
| ┣ | （图的）重数：重复次数最多的边的重复次数。 | | | | | | | | | |
| ┣ | （从到的）通路Walk（路径Path||）：顶点与边相互交错且的有限非空序列。 | | | | | | | | | |
| ┃ | ┣ | （通路的）起点：。 | | | | | | | | |
| ┃ | ┣ | （通路的）终点：。 | | | | | | | | |
| ┃ | ┣ | （通路的）长度：通路中边的数目。 | | | | | | | | |
| ┃ | ┣ | （通路的）子通路（子路径）：通路的一个连续子序列。 | | | | | | | | |
| ┃ | ┗ | 分类 | | | | | | | | |
| ┃ |  | ┃ | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | |  | | | 按起点与终点关系（！） | | | | | 闭 | | | 开 | | 通路 | | | 环（回路、路径是闭的） | | | 路径是开的 | | ┗ | 按边集与的关系（？） | |  | |  | ┗ | 巡回：经过的每边至少1次的闭通路。 |  | | ┣ | 复杂通路 | | 复杂回路：有边重复。 | | |  | | ┗ | 道路（简单通路） | | 回路Circuit（简单回路） | | |  | |  | ┗ | 路径（初级通路） | 圈Cycle（初级回路、环是简单的） | | |  | | | | | | | | |
| ┃ |  | ┣ | 按每个顶点出现次数（？） | | | | | | | |
| ┃ |  | ┃ | ┗ | 闭 赋权图最佳推销员回路：经过每个顶点至少1次的权最小的闭通路。 | | | | | | |
| ┃ |  | ┗ | 按边是否重复（？） | | | | | | | |
| ┃ |  |  | ┣ | 复杂通路：有边重复出现的通路。 | | | | | | |
| ┃ |  |  | ┗ | 道路Trail（行迹Trace、简单通路||）：边不重复的通路。 | | | | | | |
| ┃ |  |  |  | ┣ | 按每条边出现次数（？） | | | | | |
| ┃ |  |  |  | ┃ | ┗ | 欧拉道路：经过的每边正好1次的道路。 | | | | |
| ┃ |  |  |  | ┃ |  | ┗ | 闭欧拉巡回：经过的每边正好1次的闭通路。 | | | |
| ┃ |  |  |  | ┗ | 按顶点是否重复（？） | | | | | |
| ┃ |  |  |  |  | ┗ | 路径Path（轨道Track、初级通路、路径是简单的、简单路径||）：顶点不重复的道路。 | | | | |
| ┃ |  |  |  |  |  | ┣ | 赋权图（路径的）权：是赋权图中从到的路径，。 | | | |
| ┃ |  |  |  |  |  | ┣ | 赋权图（从到的）最短路：在赋权图中，从顶点到顶点的具有最小权的路。 | | | |
| ┃ |  |  |  |  |  | ┗ | 按每个顶点出现次数（？） | | | |
| ┃ |  |  |  |  |  |  | ┗ | （的）哈密顿路径（|路径）：经过的每个顶点正好1次的路径。 | | |
| ┃ |  |  |  |  |  |  |  | ┗ | 闭哈密顿圈（圈）：经过的每个顶点正好1次的圈。 | |
| ┃ |  |  |  |  |  |  |  |  | ┗ | 闭 赋权图最佳圈：权最小的哈密顿圈。 |
| ┣ | 关系 | | | | | | | | | |
| ┃ | ┣ | 顶点与顶点（？） | | | | | | | | |
| ┃ | ┃ | ┗ | （是从经过）可达的（||有向图 ）：从顶点到顶点存在1条路径。 | | | | | | | |
| ┃ | ┃ |  | ┣ | 无向图（顶点与顶点是）连通的：从顶点到顶点存在通路。规定任何顶点与自身是连通的。 | | | | | | |
| ┃ | ┃ |  | ┣ | 有向图（顶点）可达（顶点）：从顶点到顶点存在通路。规定任何顶点到自身是可达的。 | | | | | | |
| ┃ | ┃ |  | ┣ | （顶点和顶点）（关于边）相邻（、是相邻的）：有边联结2个顶点、。 | | | | | | |
| ┃ | ┃ |  | ┗ | （顶点）邻接于（顶点）（邻接到）：是图中的1条边。在无向图中，邻接关系是对称的；在有向图中，邻接关系不是对称的。 | | | | | | |
| ┃ | ┣ | 边与边（？） | | | | | | | | |
| ┃ | ┃ | ┣ | （边和边）（关于顶点）相邻（、是相邻的）：2条边、有公共顶点。 | | | | | | | |
| ┃ | ┃ | ┗ | 重边：联结同1对顶点的边。 | | | | | | | |
| ┃ | ┃ |  | ┣ | 无向图平行边：关联一对顶点的无向边多于条，这些边。 | | | | | | |
| ┃ | ┃ |  | ┗ | 有向图有向平行边（平行边）：有多于条的边始点与终点相同，这些边。 | | | | | | |
| ┃ | ┣ | 顶点与边（？） | | | | | | | | |
| ┃ | ┃ | ┣ | 互相关联：边和它的端点称为互相关联的。 | | | | | | | |
| ┃ | ┃ | ┃ | ┣ | 无向图（与顶点和顶点）关联（、为的端点）：是中的1条边。 | | | | | | |
| ┃ | ┃ | ┃ | ┣ | 有向图（）射入或进入（顶点）（、为的端点，为的终点，与顶点和顶点关联，是与顶点关联的入边）：是中的1条边。 | | | | | | |
| ┃ | ┃ | ┃ | ┗ | 有向图（）射出或离开（顶点）（、为的端点，为的始点，与顶点和顶点关联，是与顶点关联的出边）：是中的1条边。 | | | | | | |
| ┃ | ┃ | ┗ | 无向图（与顶点的）关联次数：若，；若，；若不是的端点，。 | | | | | | | |
| ┃ | ┗ | 图与图（？） | | | | | | | | |
| ┃ |  | ┗ | 无向图（与是）同构的（||）：设两个无向图，，存在双射函数，使得对于任意的当且仅当，并且和的重数相同。 | | | | | | | |
| ┣ | 分类 | | | | | | | | | |
| ┃ | ┗ | 超图：含有超边的图。 | | | | | | | | |
| ┃ |  | ┗ | 按所有超边的端点数量（？） | | | | | | | |
| ┃ |  |  | ┗ | 图 | | | | | | |
| ┃ |  |  |  | ┣ | 按顶点的数量和边的数量（！） | | | | | |
| ┃ |  |  |  | ┃ | ┣ | （图是）无穷的：图有无穷多的顶点和/或边。 | | | | |
| ┃ |  |  |  | ┃ | ┃ | ┗ | 按每个顶点的度（？） | | | |
| ┃ |  |  |  | ┃ | ┃ |  | ┗ | （图是）局部有穷的：每个顶点的度是有限的。 | | |
| ┃ |  |  |  | ┃ | ┗ | （图是）有穷的：否则。 | | | | |
| ┃ |  |  |  | ┃ |  | ┣ | 按顶点的数量（？） | | | |
| ┃ |  |  |  | ┃ |  | ┃ | ┗ | 空图：。 | | |
| ┃ |  |  |  | ┃ |  | ┣ | 按边的数量（？） | | | |
| ┃ |  |  |  | ┃ |  | ┃ | ┗ | 空图（秃图、零图）：。 | | |
| ┃ |  |  |  | ┃ |  | ┗ | 平凡图：。 | | | |
| ┃ |  |  |  | ┣ | 按自环的数量和重边的数量（！） | | | | | |
| ┃ |  |  |  | ┃ | ┣ | 多重图：有自环或有重边的图/含平行边的图。 | | | | |
| ┃ |  |  |  | ┃ | ┗ | 简单图：没有自环且没有重边的图/既不含平行边也不含自环的图。  二元组，顶点集（||、），边集（||、），。 | | | | |
| ┃ |  |  |  | ┃ |  | ┗ | 按顶点的度（？） | | | |
| ┃ |  |  |  | ┃ |  |  | ┣ | （）正则图：每个顶点的度是。 | | |
| ┃ |  |  |  | ┃ |  |  | ┃ | ┗ | （阶）完全图（完备图||）：阶为的正则图。连通图 | |
| ┃ |  |  |  | ┃ |  |  | ┣ | 无向图阶无向完全图（||）：设是阶无向简单图，中任何顶点都与其余的个顶点相邻，。 | | |
| ┃ |  |  |  | ┃ |  |  | ┗ | 有向图阶有向完全图：设是阶有向简单图，，。 | | |
| ┃ |  |  |  | ┣ | 按所有边是否有方向（！） | | | | | |
| ┃ |  |  |  | ┃ | ┣ | 混合图：既有无向边又有有向边的图。 | | | | |
| ┃ |  |  |  | ┃ | ┣ | 有向图（||）：每一条边都是有向边的图。  二元组，是一个非空的有穷集合，是卡氏积的一个多重子集。 | | | | |
| ┃ |  |  |  | ┃ | ┗ | 无向图：每一条边都是无向边的图。  二元组，是一个非空的有穷集合，是无序积的一个多重子集。 | | | | |
| ┃ |  |  |  | ┣ | 按所有两个顶点是否可达（？） | | | | | |
| ┃ |  |  |  | ┃ | ┣ | 无向图（图是）连通图（图是连通的）：图是平凡图或中任意两顶点是连通的。 | | | | |
| ┃ |  |  |  | ┃ | ┃ | ┗ | 无向图（图的）连通分量（连通分支）：是中顶点之间的连通关系。按照将划分成若干个等价类，由它们导出的导出子图。 | | | |
| ┃ |  |  |  | ┃ | ┃ |  | ┗ | （||） | | |
| ┃ |  |  |  | ┃ | ┣ | 无向图（图是）非连通图：否则。 | | | | |
| ┃ |  |  |  | ┃ | ┗ | 有向图（图是）弱连通图（|连通图）：略去中所有有向边的方向后所得的无向图是连通图。 | | | | |
| ┃ |  |  |  | ┃ |  | ┗ | 有向图（图是）单连通图：任意两顶点至少一个可达另一个。 | | | |
| ┃ |  |  |  | ┃ |  |  | ┗ | 有向图（图是）强连通图（图是强连通的）：任意两个顶点相互可达。 | | |
| ┃ |  |  |  | ┃ |  | ┣ | 按是否存在欧拉巡回（？） | | | |
| ┃ |  |  |  | ┃ |  | ┃ | ┗ | 欧拉图：存在欧拉道路的图/存在欧拉巡回的图。 | | |
| ┃ |  |  |  | ┃ |  | ┗ | 按是否存在哈密顿圈 | | | |
| ┃ |  |  |  | ┃ |  |  | ┗ | 哈密顿图（图）：含哈密顿路径的图/含哈密顿圈的图。 | | |
| ┃ |  |  |  | ┗ | 按是否赋权（？） | | | | | |
| ┃ |  |  |  |  | ┗ | 赋权图（加权图、带权图）：将图的每一条边对应一个实数。  （||（无向边或有向边）） | | | | |
| ┣ | 运算 | | | | | | | | | |
| ┃ | ┣ | （从顶点到顶点的）距离：若从顶点出发到顶点的最短通路存在，此通路的长度；否则，无穷（）。 | | | | | | | | |
| ┃ | ┣ | 赋权图（顶点到顶点的）最短路径：从到的所有通路中权最小的通路。 | | | | | | | | |
| ┃ | ┃ | ┗ | 赋权图（顶点到顶点的）距离：到的最短路径的权。 | | | | | | | |
| ┃ | ┣ | （是的）子图（是的母图||）：，，，。 | | | | | | | | |
| ┃ | ┃ | ┣ | 赋权图（的）权（||）：。 | | | | | | | |
| ┃ | ┃ | ┣ | （是的）真子图：。 | | | | | | | |
| ┃ | ┃ | ┣ | （是的）导出子图：。 | | | | | | | |
| ┃ | ┃ | ┣ | （是的）生成子图（支撑子图）：。 | | | | | | | |
| ┃ | ┃ | ┃ | ┗ | 生成树：把一个连通图中的一些边删除得到的树。 | | | | | | |
| ┃ | ┃ | ┃ |  | ┣ | 赋权图 连通图最小生成树：一个图的权最小的生成树。 | | | | | |
| ┃ | ┃ | ┃ |  | ┗ | 赋权图 连通图最大生成树：一个图的权最大的生成树。 | | | | | |
| ┃ | ┃ | ┣ | （的）由导出的子图（关于的导出子图||）：，〡，。 | | | | | | | |
| ┃ | ┃ | ┗ | （的）由导出的子图（||）：，〡，。不含有孤立点。 | | | | | | | |
| ┃ | ┣ | 简单图（的）补图：，。 | | | | | | | | |
| ┃ | ┣ | 无向图 简单图（的）补图（||）：设是阶无向简单图，，。 | | | | | | | | |
| ┃ | ┣ | 有向图 简单图（的）补图（||）：设是阶有向简单图，，。 | | | | | | | | |
| ┃ | ┣ | 无向图删除（||）：，从中删除中的所有顶点及其关联的边。 | | | | | | | | |
| ┃ | ┃ | ┗ | （的）点割集：，，。 | | | | | | | |
| ┃ | ┃ |  | ┗ | 割点：若点割集中只有一个顶点，。 | | | | | | |
| ┃ | ┗ | 无向图删除（||）：，从中删除中的所有边。 | | | | | | | | |
| ┃ |  | ┗ | （的）边割集（|割集）：，，。 | | | | | | | |
| ┃ |  |  | ┗ | 割边（桥）：若边割集中只有一条边，。 | | | | | | |
| ┗ | 性质 | | | | | | | | | |
|  | ┣ | 对偶图：平面图的对偶图是平面图。  　　　　平面连通图的对偶图的对偶图是原图。 | | | | | | | | |

|  |  |  |
| --- | --- | --- |
|  | ┣ | 平面图（的）对偶图（||）：原图的每个面（包括外部面）得到一个顶点，[每对面得到的一对顶点]得到[两面的公共边数]条边（一条非自环的边若只属于一个面，得到一个自环；一个自环得到一条非自环的边）。 |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ┃ |  |  |  | ┃ |  | ┗ | 按所有由点集的子集导出的子图的边的数量（？） | | | |
| ┃ |  |  |  | ┃ |  |  | ┗ | 二元图：。 | | |
| ┃ |  |  |  | ┃ |  |  |  | ┗ | 按边的数量（？） | |
| ┃ |  |  |  | ┃ |  |  |  |  | ┗ | 完备二元图（||（、为、的顶点的数目））：。连通图 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ┃ |  |  |  | ┣ | 无向图 简单图按环的数量（？） | | | | | | | |
| ┃ |  |  |  | ┃ | ┗ | 森林：无环。 | | | | | | |
| ┃ |  |  |  | ┃ |  | ┗ | 按是否连通（？） | | | | | |
| ┃ |  |  |  | ┃ |  |  | ┗ | 树：连通。 | | | | |
| ┃ |  |  |  | ┃ |  |  |  |  | 有根树：在树中指定一个节点为根节点。 | | | |
| ┃ |  |  |  | ┃ |  |  |  |  |  | 层 | | |
| ┃ |  |  |  | ┃ |  |  |  |  |  | 高度 | | |
| ┃ |  |  |  | ┃ |  |  |  |  |  | 宽度 | | |
| ┃ |  |  |  | ┃ |  |  |  |  |  | （节点的）祖先 | | |
| ┃ |  |  |  | ┃ |  |  |  |  |  | ┗ | （节点的）父节点（父亲、双亲、双亲节点） | |
| ┃ |  |  |  | ┃ |  |  |  |  |  | （节点的）子孙 | | |
| ┃ |  |  |  | ┃ |  |  |  |  |  | ┗ | （节点的）子节点（孩子、儿子、子女） | |
| ┃ |  |  |  | ┃ |  |  |  |  |  | 叶节点（叶子节点） | | |
| ┃ |  |  |  | ┃ |  |  |  |  |  | 以节点为根的树 | | |
| ┃ |  |  |  | ┃ |  |  |  |  |  | （节点的）子树 | | |
| ┃ |  |  |  | ┃ |  |  |  |  |  | 按子树数量（？） | | |
| ┃ |  |  |  | ┃ |  |  |  |  |  | ┗ | 叉树：每个节点的子树不超过个 | |
| ┃ |  |  |  | ┃ |  |  |  |  |  |  |  | 二叉树： |
| ┃ |  |  |  | ┃ |  |  |  |  |  |  | | |
| ┃ |  |  |  | ┣ | 按是否能展在平面上（！） | | | | | | | |
| ┃ |  |  |  |  | ┣ | 平面图：可以画在平面上并且可以使得不同的边互不交叠的图。 | | | | | | |
| ┃ |  |  |  |  | ┃ | ┣ | （图的）外部面：图的外面的区域。 | | | | | |
| ┃ |  |  |  |  | ┃ | ┗ | （图的）内部面：图里面每个被顶点和边分割出来的封闭并连通的区域。 | | | | | |
| ┃ |  |  |  |  | ┗ | 非平面图：否则。 | | | | | | |
| ┃ |  |  |  |  |  |  |  | | | | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |