

作者	生姜 DrGinger
脚本	生姜 DrGinger
视频	崔崔 CuiCui
开源学习资源	<a href="https://github.com/Visualize-ML">https://github.com/Visualize-ML</a>
平台	<a href="https://www.youtube.com/@DrGinger_Jiang">https://www.youtube.com/@DrGinger_Jiang</a> <a href="https://space.bilibili.com/3546865719052873">https://space.bilibili.com/3546865719052873</a> <a href="https://space.bilibili.com/513194466">https://space.bilibili.com/513194466</a>

## 15.2 有向图



### 本节你将掌握的核心技能：

- ▶ 有向图是由节点和有向边组成的结构。
- ▶ 有向边有方向，表示从一个节点“指向”另一个节点的单向关系。
- ▶ 有向图的邻接矩阵是一个方阵，矩阵第  $i$  行第  $j$  列的元素表示从节点  $i$  指向节点  $j$  的边的数量或权重。
- ▶ 有向图邻接矩阵的  $k$  次幂的第  $i$  行第  $j$  列元素表示从节点  $i$  到节点  $j$  恰好经过  $k$  条边的路径数量。
- ▶ 有向图中，出度是节点发出的边数，对应邻接矩阵每行元素之和；入度是指向该节点的边数，对应每列元素之和。

本节讲解有向图，本节和上一节无向图知识点结构类似，建议大家平行阅读。

### 有向图

图论中，**有向图** (directed graph) 是**节点** (node)、**有向边** (directed edge) 构成的数据结构。

上一节提过，无向图的边体现的是节点之间的连接关系；而有向图的有向边用来表示“从哪儿到哪儿”的单向关系。

举个例子，如图 1 所示，假设我们有 4 个城市，每个城市可以看作一个节点。图 1 这幅图一共有四个节点： $a$ 、 $b$ 、 $c$ 、 $d$ 。

城市之间若存在航班，就画一条有向线段连接，这条线段就是**有向边**。

比如，有一班航班从  $a$  飞往  $b$ ，我们就在图中画一条从  $a$  指向  $b$  的箭头。于此同时，也有航班从  $b$  飞往  $a$ ，再画一条从  $b$  指向  $a$  的箭头。

观察图 1，我们发现只有  $a$ 、 $b$  和  $c$ 、 $d$  城市之间存在往返航班。

有些城市之间没有直飞，比如，要从  $c$  飞到  $a$  就很麻烦，需要走  $c \rightarrow d \rightarrow a$  这条线路。

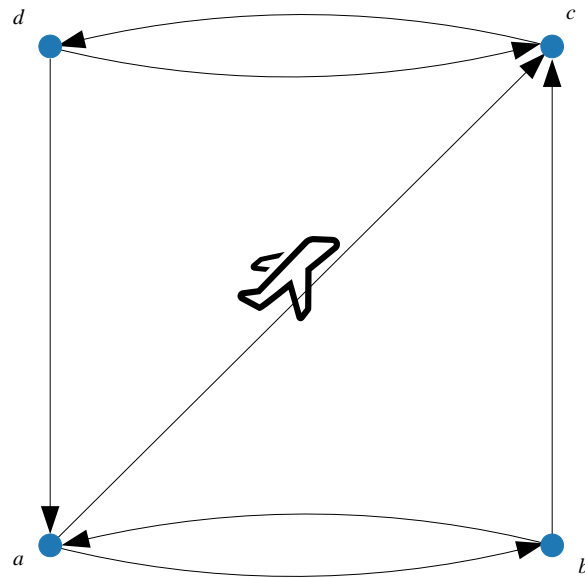


图 1.4 个城市航班构成的有向图

## NetworkX 绘制有向图

代码 1 用 NetworkX 绘制图 1 有向图。下面聊聊其中关键语句。

**a** 用 `nx.DiGraph()` 创建了一个有向图的对象，命名为 `directed_G`。有向图的意思是边有方向，比如从 `a` 到 `b` 是一条（有向）边，而从 `b` 到 `a` 是另一条（有向）边，这两个方向是区分开的。`DiGraph` 是“Directed Graph”的缩写。

**b** 用 `add_nodes_from()` 往图里一次性添加四个节点，它们的名字分别是 '`a`'、'`b`'、'`c`'、'`d`'。这些名字可以随便起，代表我们图里的元素，比如城市。

**c** 往图里添加有向边。我们一次性添加了七条边，每条边都是一个从某个节点出发，指向另一个节点的“箭头”。比如 `(a,b)` 表示从节点 '`a`' 指向 '`b`' 的一条边。注意这里同时有 `(a,b)` 和 `(b,a)`，表示它们之间是双向连接的，两边都有航班或关系。`add_edges_from` 是一次添加多条边的函数。


**d** 定义每个节点在图上应该显示在哪个位置。我们人为地给 '`a`'、'`b`'、'`c`'、'`d`' 四个节点设置了二维坐标。

**e** 用来可视化这副有向图。用 `draw_networkx` 把图 `directed_G` 画出来。

`pos=pos` 表示使用我们刚刚指定的节点坐标。

`node_size=188` 设置每个节点的大小，数字越大，圆圈越大。

`connectionstyle='arc3, rad = 0.1'` 是一个小技巧，用来把双向边画成弯曲的弧线，而不是两条直线重叠。比如 '`a`' 到 '`b`' 和 '`b`' 到 '`a`' 同时存在时，这个参数可以让它们弯曲显示，避免看不清。

代码 1. 用 NetworkX 绘制有向图 |  LA\_15\_02\_01.ipynb

```

## 初始化
import matplotlib.pyplot as plt
import networkx as nx

## 有向图
a directed_G = nx.DiGraph()

## 节点
b directed_G.add_nodes_from(['a', 'b', 'c', 'd'])

## 有向边
c directed_G.add_edges_from([('a', 'b'),
                             ('b', 'a'),
                             ('a', 'c'),
                             ('d', 'a'),
                             ('b', 'c'),
                             ('c', 'd'),
                             ('d', 'c')])

## 定义节点坐标
d pos = {'a': [0, 0],
         'b': [1, 0],
         'c': [1, 1],
         'd': [0, 1]}

## 可视化
e plt.figure(figsize = (6,6))
nx.draw_networkx(directed_G, pos = pos, node_size = 188,
                 connectionstyle = 'arc3, rad = 0.1')

```

## 邻接矩阵

类似图 1 的有向图也都可以通过邻接矩阵来表达。

图 1 有向图的邻接矩阵为

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \quad (1)$$

简单来说，有向图的邻接矩阵是一个方阵，其中第  $i$  行第  $j$  列的元素表示从节点  $i$  指向节点  $j$  的边的数量或权重（有权无向图）。如图 1(a) 所示，存在节点  $a$  指向  $b$  的边，对应邻接矩阵  $A$  的第 1 行、第 2 列元素为 1。

**!** 注意，先行、再列这个先后顺序。

上一节介绍过，无向图的邻接矩阵都是对称矩阵；与之相反，有向图的邻接矩阵一般都不是对称矩阵。

对于有向图，当且仅当图中每一条有向边的反向边也同时存在时，有向图的邻接矩阵是对称矩阵。换句话说，就是图中任意一对节点之间，如果存在从节点  $a$  到节点  $b$  的边，也必须存在从节点  $b$  到节点  $a$  的边，整个图才会对应一个对称的邻接矩阵。此时，图虽然是有向图，但结构上与无向图没有本质区别。

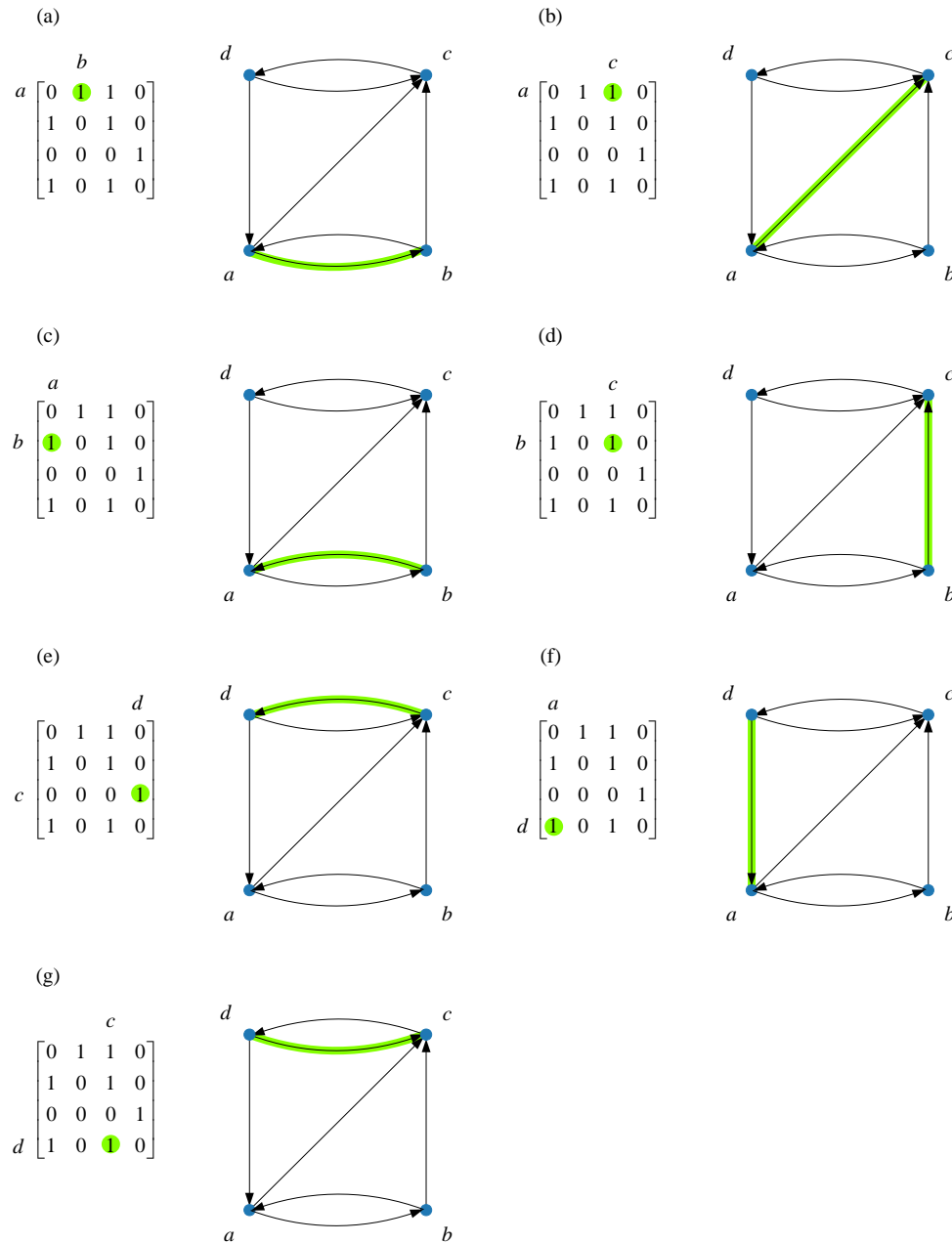


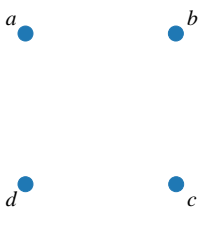
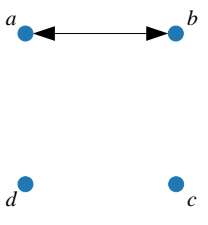
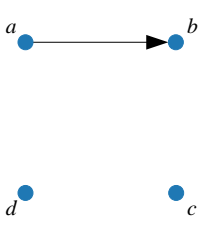
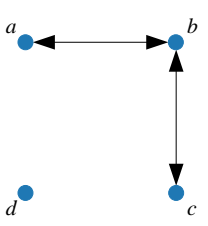
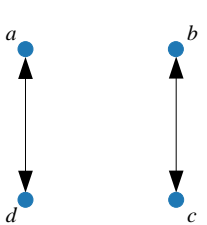
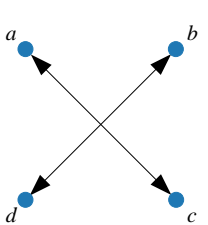
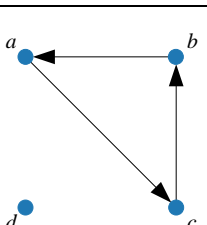
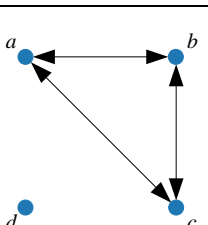
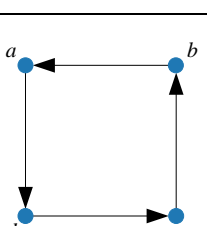
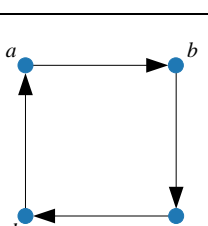
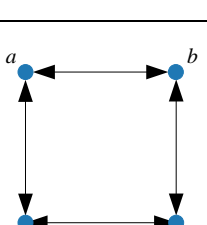
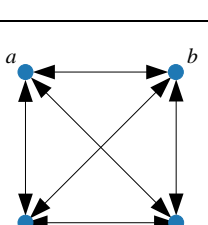
图 2. 有向图邻接矩阵的每个元素



LA\_15\_02\_02.ipynb 用邻接矩阵 (二维数组) 构造有向图，请大家自学。

? 请大家逐一分析表 1 中每个有向图和邻接矩阵。

表 1. 4 个节点构造的几个有向图及邻接矩阵

有向图	邻接矩阵	有向图	邻接矩阵
	$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$		$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$
	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$		$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$
	$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$		$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$
	$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$		$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$
	$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$		$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$
	$\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$		$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$

## 再看邻接矩阵

如图 3 所示，邻接矩阵  $A$  的第 1 行数值为 1 的元素代表  $a$  可以单向直达的节点。也就是说，不途径任何节点，节点  $a$  可以直接走到  $b$ 、 $c$ 。可以写成如下矩阵乘法

$$[1 \ 0 \ 0 \ 0] @ A = [1 \ 0 \ 0 \ 0] @ \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} = [0 \ 1 \ 1 \ 0] \quad (2)$$

上式相当于取出矩阵  $A$  的第一行。

? 请大家分析邻接矩阵  $A$  的剩余行。

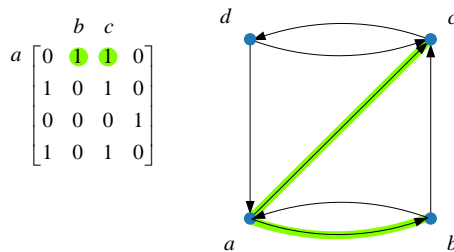


图 3.  $a$  直达两个节点，有向图邻接矩阵  $A$  的第 1 行

如图 4 所示，邻接矩阵  $A$  的第 1 列展示的是直达  $a$  的两个节点，对应如下乘法

$$A @ \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} @ \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \quad (3)$$

上式相当于取出矩阵  $A$  的第一列。

? 请大家用相同的思路分析邻接矩阵  $A$  剩余的列。

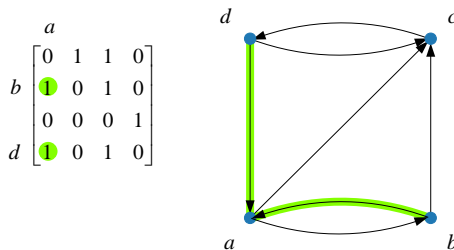


图 4. 直达  $a$  的两个节点，有向图邻接矩阵  $A$  的第 1 列

节点  $a$  途径一个节点到达  $c$  的路径数量可以通过以下矩阵乘法得到

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微视频均发布在 B 站——生姜 DrGinger：<https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：[jiang.visualize.ml@gmail.com](mailto:jiang.visualize.ml@gmail.com)

$$A @ A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} @ \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \quad (4)$$

如图 5 所示， $A @ A$  结果第 1 行、第 3 列元素值就是节点  $a$  途径一个节点到达  $c$  的路径数量。

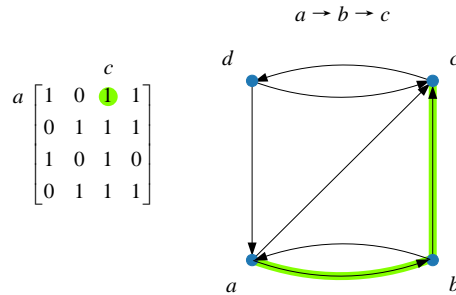


图 5. 节点  $a$  途径一个节点到达  $c$  的路径

如图 6 所示， $A @ A$  结果第 3 行、第 1 列元素值就是节点  $c$  途径一个节点到达  $a$  的路径数量。

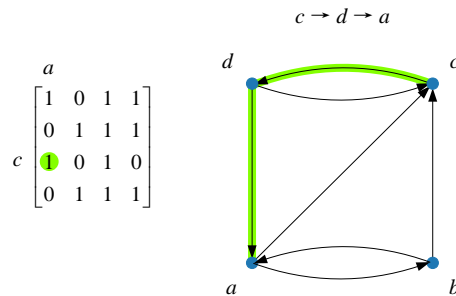


图 6. 节点  $c$  途径一个节点到达  $a$  的路径

**?** 请大家分析 (4) 中  $A @ A$  结果中剩余元素的含义。

要计算从  $a$  到  $c$  途径不超过一个节点的路径数量，我们可以利用如下运算

$$A + A @ A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 2 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 2 & 1 \end{bmatrix} \quad (5)$$

请大家自行分析如上结果。

## 度、度矩阵

和无向图一样，有向图任意一个节点的度 (degree) 是与它相连的边的数量。

但是，有向图中由于边有方向，我们更关心入度 (indegree)、出度 (outdegree) 这两个概念。

在有向图中，节点的入度是指指向该节点的边的数量，即从其他节点指向该节点的有向边的数量。而出度是指从该节点出发的边的数量，即从该节点指向其他节点的有向边的数量。这两个概念用于描述有向图中节点的连接性质，入度和出度的总和等于节点的度数。

无向图邻接矩阵  $A$  沿列求和的结果就是每个节点的入度，即

$$\mathbf{1}^T \mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 3 & 1 \end{bmatrix} \quad (6)$$

入度矩阵为对角方阵

$$\begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

此外，有向图邻接矩阵  $A$  的格拉姆矩阵  $A^T @ A$  的主对角线元素也是入度值

$$\mathbf{A}^T \mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}^T @ \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 2 & 0 \\ 0 & 1 & 1 & 0 \\ 2 & 1 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

图 7 所示为有向图四个节点的入度。

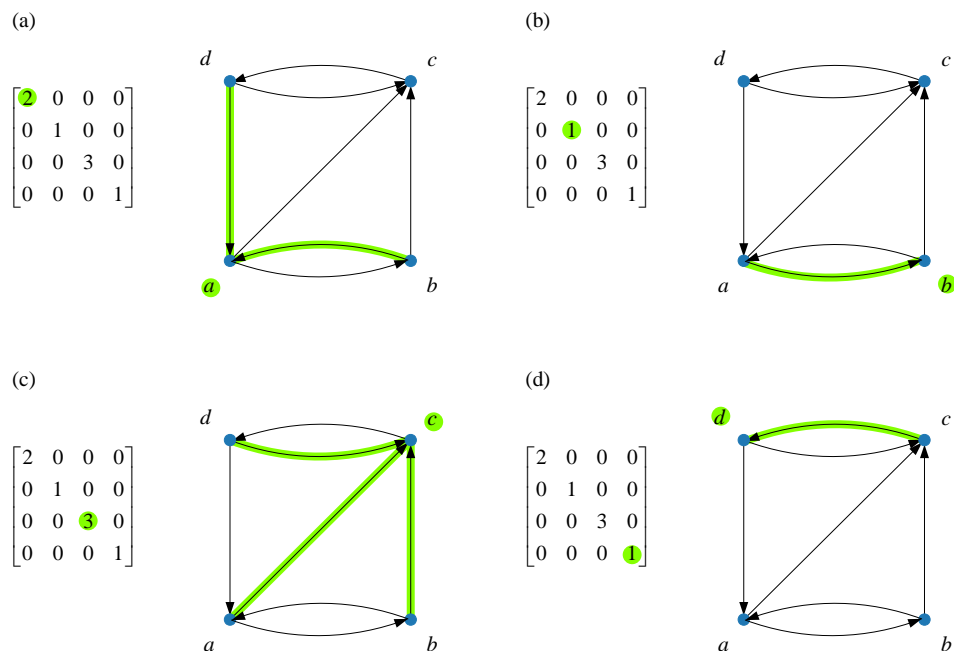




图 7. 有向图四个节点的入度

无向图邻接矩阵  $A$  沿行求和的结果就是每个节点的出度，即

$$AI = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 1 \\ 2 \end{bmatrix} \quad (9)$$

出度矩阵也是对角方阵

$$\begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} \quad (10)$$

有向图邻接矩阵  $A$  转置的格拉姆矩阵  $AA^T$  的主对角线元素也是出度值

$$AA^T = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} @ \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}^T = \begin{bmatrix} 2 & 1 & 0 & 1 \\ 1 & 2 & 0 & 2 \\ 0 & 0 & 1 & 0 \\ 1 & 2 & 0 & 2 \end{bmatrix} \quad (11)$$

图 8 所示为有向图四个节点的出度。

⚠ 注意，用格拉姆矩阵计算出度、入度矩阵仅仅适用于无权有向图。

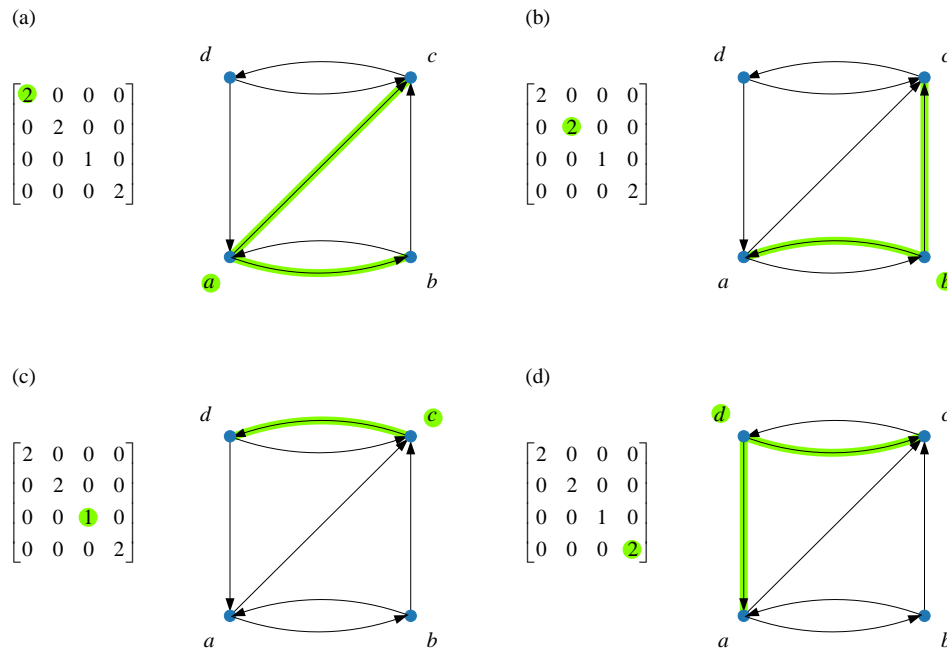


图 8. 有向图四个节点的出度

## 传球问题

邻接矩阵可以用来解决很多有趣的数学问题，比如传球问题。



传球问题这个例子来自《数据有道》。

有  $a$ 、 $b$ 、 $c$ 、 $d$ 、 $e$ 、 $f$  六名同学相互之间传球一只球。规则是，某个人每次传球可以传给其他任何人，但是不能传给自己。从  $a$  开始传球，传球 4 次，球最终回到  $a$  的手中，请大家计算一共有多少种传法。

图 9 所示为一种传法，传球路线为  $a \rightarrow f \rightarrow e \rightarrow b \rightarrow a$ 。

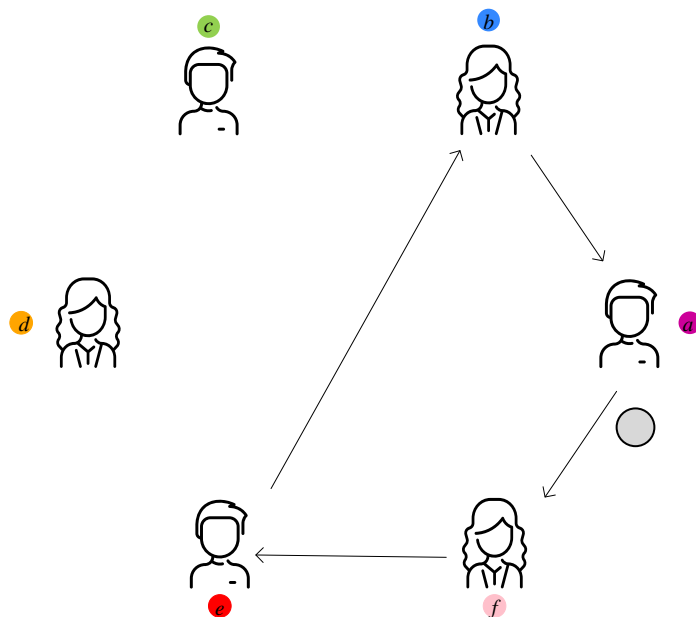


图 9. 一种传法

图 10 展示传球 4 次的所有路径，我们需要找到从  $a$  出发再回到  $a$  的所有传球线路。

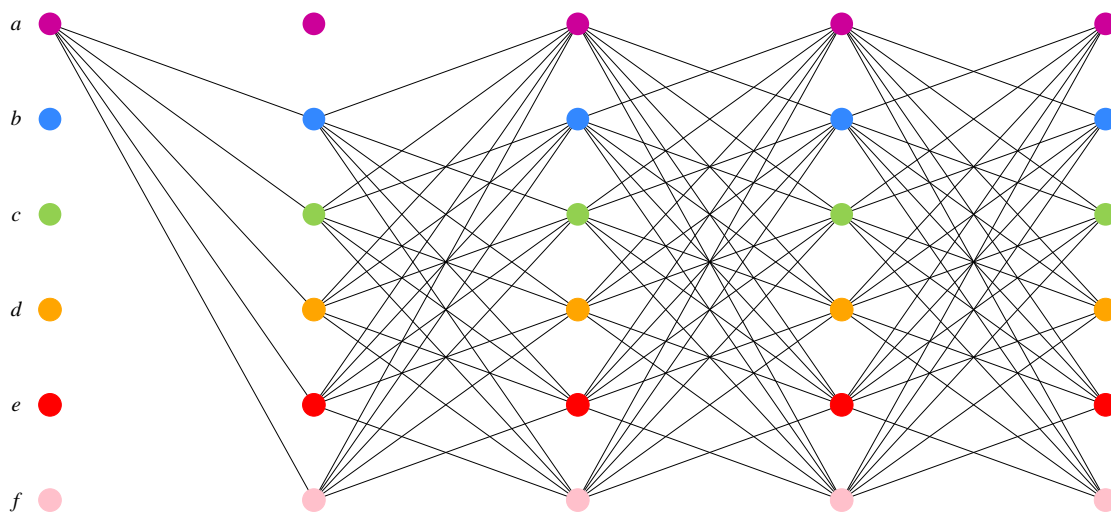


图 10. 所有可能路径的网络

把  $a$ 、 $b$ 、 $c$ 、 $d$ 、 $e$ 、 $f$  六名同学看成是六个节点的话，他们之间的传球关系可以抽象成图 11 所示有向图。而这幅有向图的邻接矩阵  $A$  为

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger：<https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：[jiang.visualize.ml@gmail.com](mailto:jiang.visualize.ml@gmail.com)

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \quad (12)$$

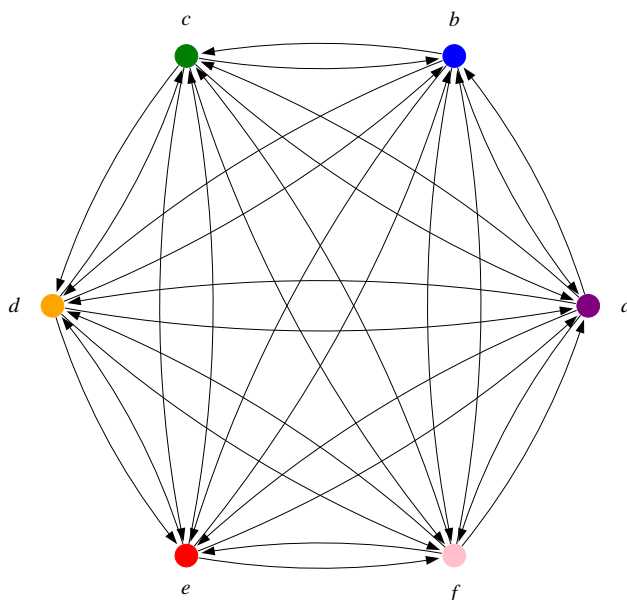


图 11. 代表传球问题的有向图

下面聊聊如何利用邻接矩阵  $A$  求解这个传球问题。

### 第 1 次传球

球最开始在  $a$  同学手里，将这个状态写成  $x_0$

$$x_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (13)$$

而矩阵乘法  $Ax_0$  代表， $a$  同学手里的球在第 1 次传球后几种路径，具体结果为

$$x_1 = Ax_0 = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} @ \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad (14)$$

如图 12 所示，这个结果表示，经过一次传球后，球可以在除了  $a$  之外的另外五名同学手上，也就是五种路径。这也是第 2 次传球的起点。

将向量  $\mathbf{x}_1$  的所有元素求和结果为 5。这个 5 实际上代表了  $5^1$ ，相当于一次传球后“一生五”。

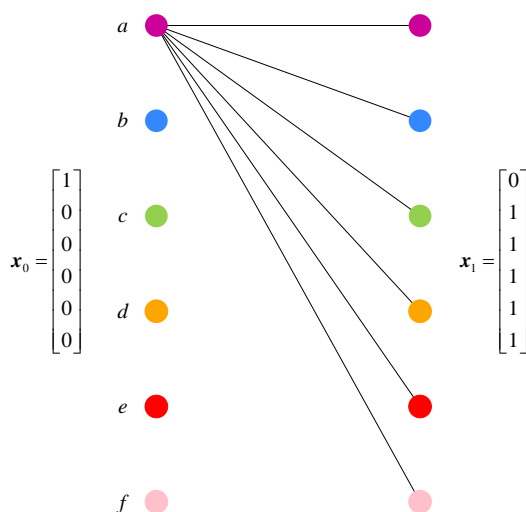


图 12. 矩阵乘法  $\mathbf{x}_1 = \mathbf{A}\mathbf{x}_0$  代表的具体含义

矩阵  $\mathbf{A}$  所有元素求和的结果为 30，即  $6 \times 5$  (6 代表 6 个节点，5 代表每个节点有 5 条路径)。图 13 展示了这 30 条路径。

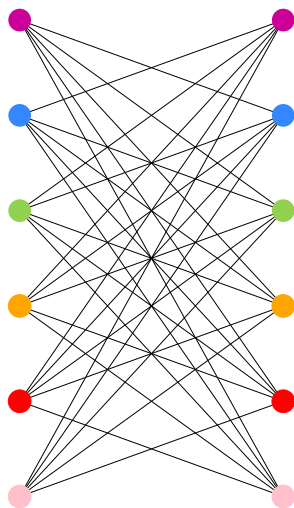


图 13. 矩阵  $\mathbf{A}$  所有元素求和

## 第 2 次传球

如图 14 所示，矩阵乘法  $\mathbf{A}\mathbf{x}_1$  代表， $a$  同学手里的球在第 2 次传球后几种路径，具体结果为

$$\mathbf{x}_2 = \mathbf{A}\mathbf{x}_1 = \mathbf{A}\mathbf{A}\mathbf{x}_0 = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} @ \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 4 \\ 4 \\ 4 \\ 4 \\ 4 \end{bmatrix} \quad (15)$$

举个例子，向量  $\mathbf{x}_2$  的第 1 个元素为 5，这代表着 2 次传球后球回到  $a$  手上有 5 条路径。类似地，向量  $\mathbf{x}_2$  的第 2 个元素为 4，这代表着 2 次传球后球回到  $b$  手上有 4 条路径。

向量  $\mathbf{x}_2$  的所有元素求和结果为 25，代表了  $5^2$ ，相当于 2 次传球后“一生五、五生二十五”。

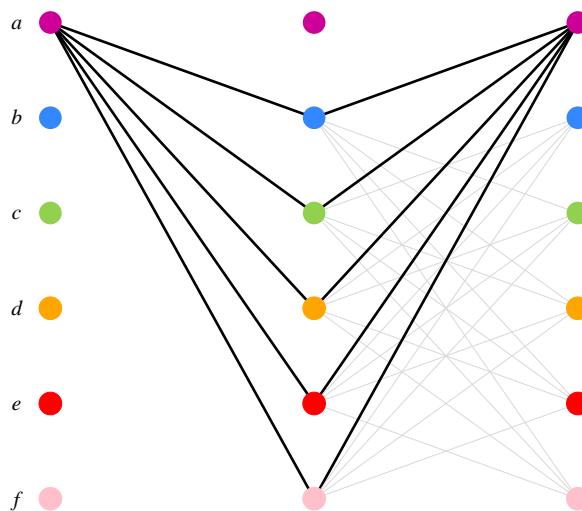


图 14. 矩阵乘法  $\mathbf{x}_2 = \mathbf{A}\mathbf{x}_1$  代表的具体含义

细心的读者可能已经发现，(15) 中核心运算是方阵  $\mathbf{A}$  的幂，即  $\mathbf{A}^2$ 。而  $\mathbf{A}^2$  的结果具体为

$$\mathbf{A}^2 = \mathbf{A}\mathbf{A} = \begin{bmatrix} 5 & 4 & 4 & 4 & 4 & 4 \\ 4 & 5 & 4 & 4 & 4 & 4 \\ 4 & 4 & 5 & 4 & 4 & 4 \\ 4 & 4 & 4 & 5 & 4 & 4 \\ 4 & 4 & 4 & 4 & 5 & 4 \\ 4 & 4 & 4 & 4 & 4 & 5 \end{bmatrix} \quad (16)$$

而 (15) 仅仅是取出  $\mathbf{A}^2$  结果的第 1 列。

换个角度，如果修改本节题目，将初始持球者换成其他同学，我们仅仅需要修改初始状态向量  $\mathbf{x}_0$

$$\mathbf{x}_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (17)$$

而对于不同初始状态向量  $\mathbf{x}_0$ ， $\mathbf{A}^2\mathbf{x}_0$  运算结果就是提取  $\mathbf{A}^2$  的不同列。

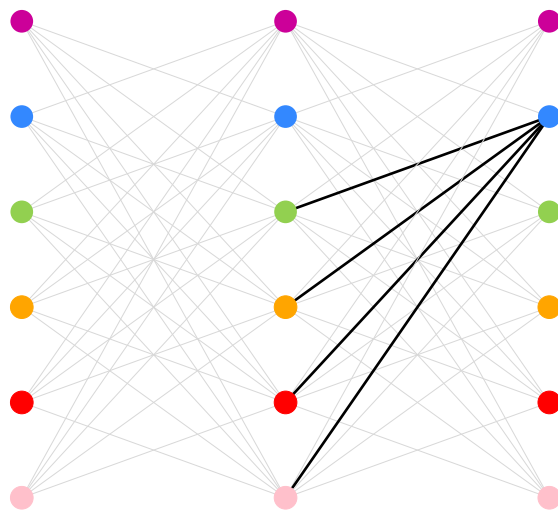
$\mathbf{A}^2$  结果也很值得细看！

$\mathbf{A}^2$  的主对角线都是 5，这代表着经过两次传球，从某位同学手中再回到本人的路径。

除了主对角线元素之外， $\mathbf{A}^2$  其他元素都是 4。出现这个结果也不意外。

举个例子，开始时如果球在  $a$  手中，两次传球后球在  $b$  手中有 4 种路径。由于  $b$  不能传给自己，这刨除一条路径。此外， $a$  不能传给自己，然后再传给  $b$ ，这又刨除了一条路径。实际上，这是利用组合数求解这个问题的内核。

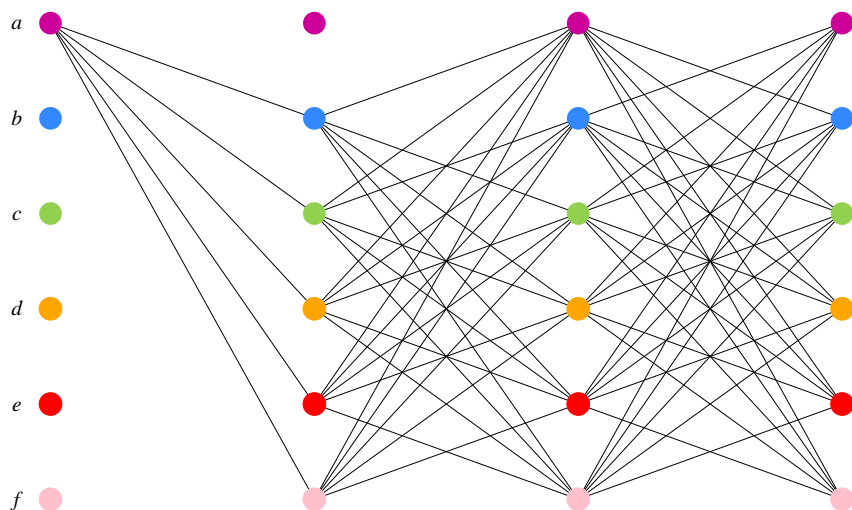
而  $\mathbf{A}^2$  的所有元素之和为 150，即  $6 \times 5 \times 5$ 。

图 15. 方阵乘幂  $A^2$  代表的具体含义

### 第 3 次传球

如图 16 所示，矩阵乘法  $Ax_2$  代表， $a$  同学手里的球在第 3 次传球后几种路径，具体结果为

$$x_3 = Ax_2 = AAAx_0 = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} @ \begin{bmatrix} 5 \\ 4 \\ 4 \\ 4 \\ 4 \\ 4 \end{bmatrix} = \begin{bmatrix} 20 \\ 21 \\ 21 \\ 21 \\ 21 \\ 21 \end{bmatrix} \quad (18)$$

图 16. 矩阵乘法  $x_3 = Ax_2$  代表的具体含义

而  $A^3$  的结果具体为

$$A^3 = AAA = \begin{bmatrix} 20 & 21 & 21 & 21 & 21 & 21 \\ 21 & 20 & 21 & 21 & 21 & 21 \\ 21 & 21 & 20 & 21 & 21 & 21 \\ 21 & 21 & 21 & 20 & 21 & 21 \\ 21 & 21 & 21 & 21 & 20 & 21 \\ 21 & 21 & 21 & 21 & 21 & 20 \end{bmatrix} \quad (19)$$

(18) 相当于取出 (19) 的第 1 列。

请大家自行分析为什么  $A^3$  的主对角线元素为 20，而其他元素为 21。

#### 第 4 次传球


矩阵乘法  $Ax_3$  代表， $a$  同学手里的球在第 4 次传球后几种路径，具体结果为

$$x_4 = Ax_3 = A^4 x_0 = \begin{bmatrix} 105 \\ 104 \\ 104 \\ 104 \\ 104 \\ 104 \end{bmatrix} \quad (20)$$

上式告诉我们本节最开始提出的问题答案为 105。

而  $A^4$  的结果具体为

$$A^4 = \begin{bmatrix} 105 & 104 & 104 & 104 & 104 & 104 \\ 104 & 105 & 104 & 104 & 104 & 104 \\ 104 & 104 & 105 & 104 & 104 & 104 \\ 104 & 104 & 104 & 105 & 104 & 104 \\ 104 & 104 & 104 & 104 & 105 & 104 \\ 104 & 104 & 104 & 104 & 104 & 105 \end{bmatrix} \quad (21)$$

 请大家思考，如果传球不超过四次，从  $a$  开始传球，球最终回到  $f$  手中，共有多少种传法。



请大家用 DeepSeek/ChatGPT 等工具完成本节如下习题。

**Q1.** 请用 NetworkX 逐一构建并绘制表 1 中所有有向图。

**Q2.** 请计算表 1 中所有有向图的邻接矩阵。

**Q3.** 请计算表 1 中所有有向图节点的出度、入度。

**Q4.** 4 个同学相互传球。规则仍然是，某个人每次传球可以传给其他任何人，但是不能传给自己。从  $a$  开始传球，传球不超过 3 次，球最终回到  $a$  的手中，请大家计算一共有多少种传法。

**Q5.** 请用 NetworkX 绘制图 11 这幅有向图，并且计算其邻接矩阵。