极客大学算法训练营
图

覃超

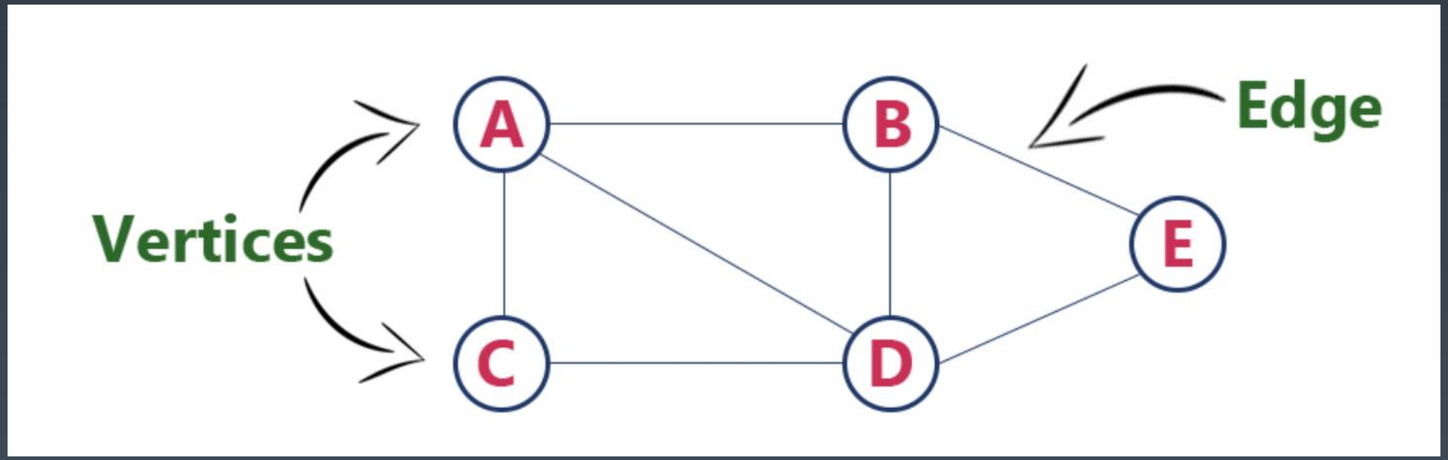
Sophon Tech 创始人，前 Facebook 工程师

目录

- 图的属性和分类
- 基于图相关的算法

什么是图?

图



图的属性

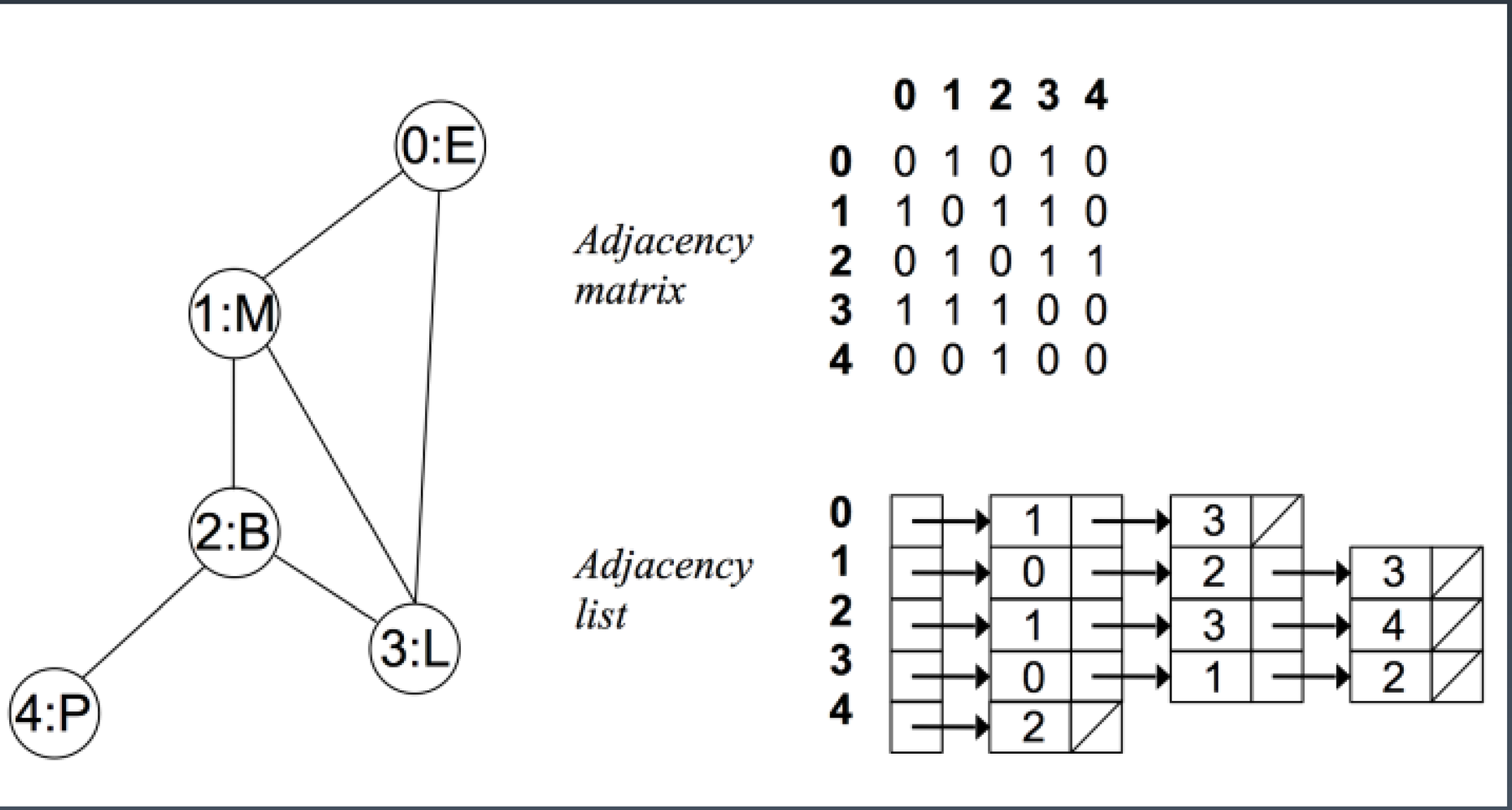
- Graph(V，E)

- V - vertex: 点
  1. 度 - 入度和出度
  2. 点与点之间：连通与否

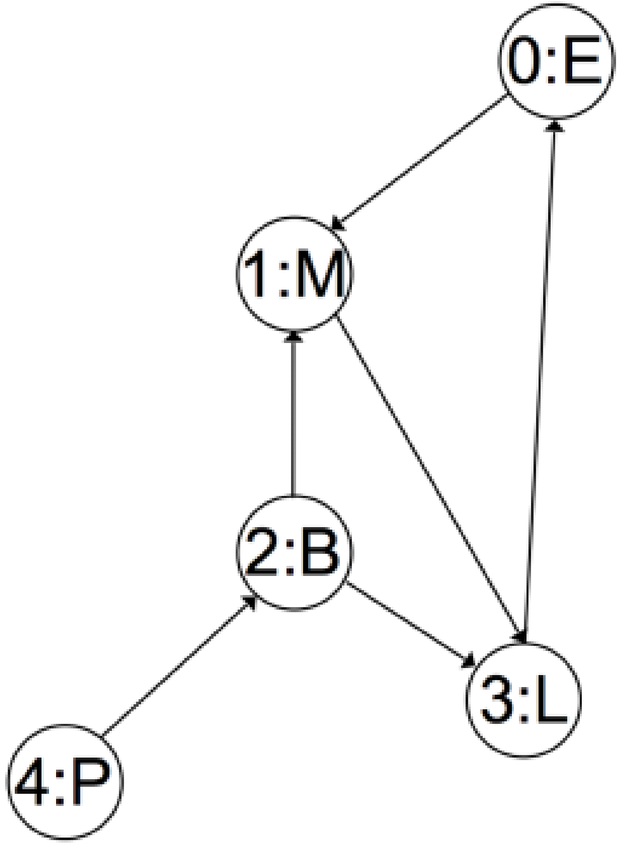
- E - edge: 边
  1. 有向和无向（单行线）
  2. 权重（边长）

图的表示和分类

图：无向无权图


0:E
1:M
2:B
3:L
4:P
Adjacency matrix
0 1 2 3 4
0 0 1 0 1 0
1 1 0 1 1 0
2 0 1 0 1 1
3 1 1 1 0 0
4 0 0 1 0 0
Adjacency list
0
1
2
3
4
1
3
0
2
3
1
0
1
3
4
2
2

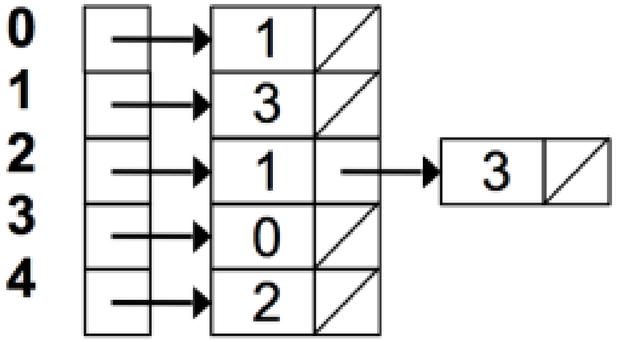
图：有向无权图



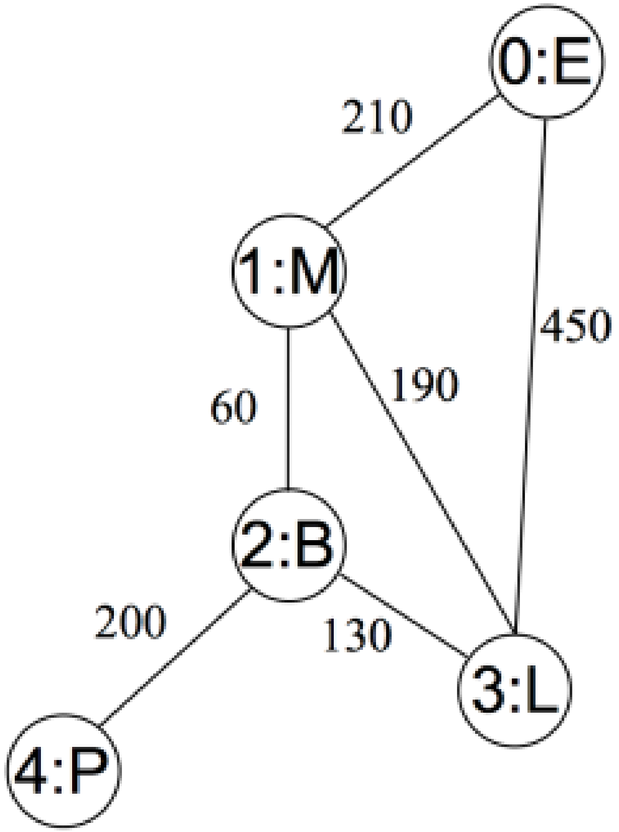
Adjacency matrix

	0	1	2	3	4
0	0	1	0	0	0
1	0	0	0	1	0
2	0	1	0	1	0
3	1	0	0	0	0
4	0	0	1	0	0

Adjacency list

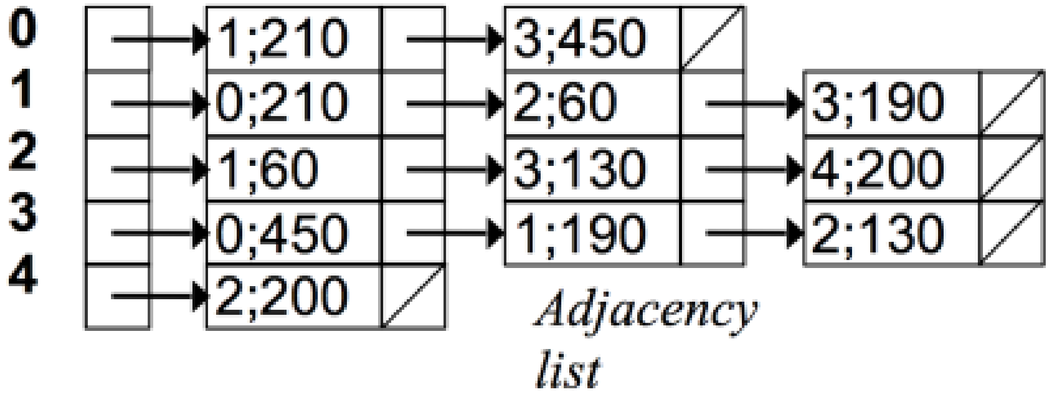


图：无向有权图



	0	1	2	3	4
0	0	210	0	450	0
1	210	0	60	190	0
2	0	60	0	130	200
3	450	190	130	0	0
4	0	0	200	0	0

Adjacency matrix



Adjacency list

基于图的常见算法

DFS 代码 - 递归写法

```
visited = set()  # 和树中的DFS最大区别

def dfs(node, visited):
    if node in visited: # terminator
        # already visited
        return

    visited.add(node)

    # process current node here.
    ...
    for next_node in node.children():
        if not next_node in visited:
            dfs(next_node, visited)
```

BFS 代码

```
def BFS(graph, start, end):

    queue = []
    queue.append([start])

    visited = set() # 和树中的BFS的最大区别

    while queue:
        node = queue.pop()
        visited.add(node)

        process(node)
        nodes = generate_related_nodes(node)
        queue.push(nodes)
```

图的高级算法

1. 连通图个数：
   https://leetcode-cn.com/problems/number-of-islands/

2. 拓扑排序（Topological Sorting）
   https://zhuanlan.zhihu.com/p/34871092

3. 最短路径（Shortest Path）：
   Dijkstra
   https://www.bilibili.com/video/av25829980?from=search&seid=13391343514095937158

4. 最小生成树（Minimum Spanning Tree）：
   https://www.bilibili.com/video/av84820276?from=search&seid=17476598104352152051

THANKS!