

极客大学算法训练营

第4期

直播课一

覃超

Sophon Tech 创始人，前 Facebook 工程师

空间复杂度

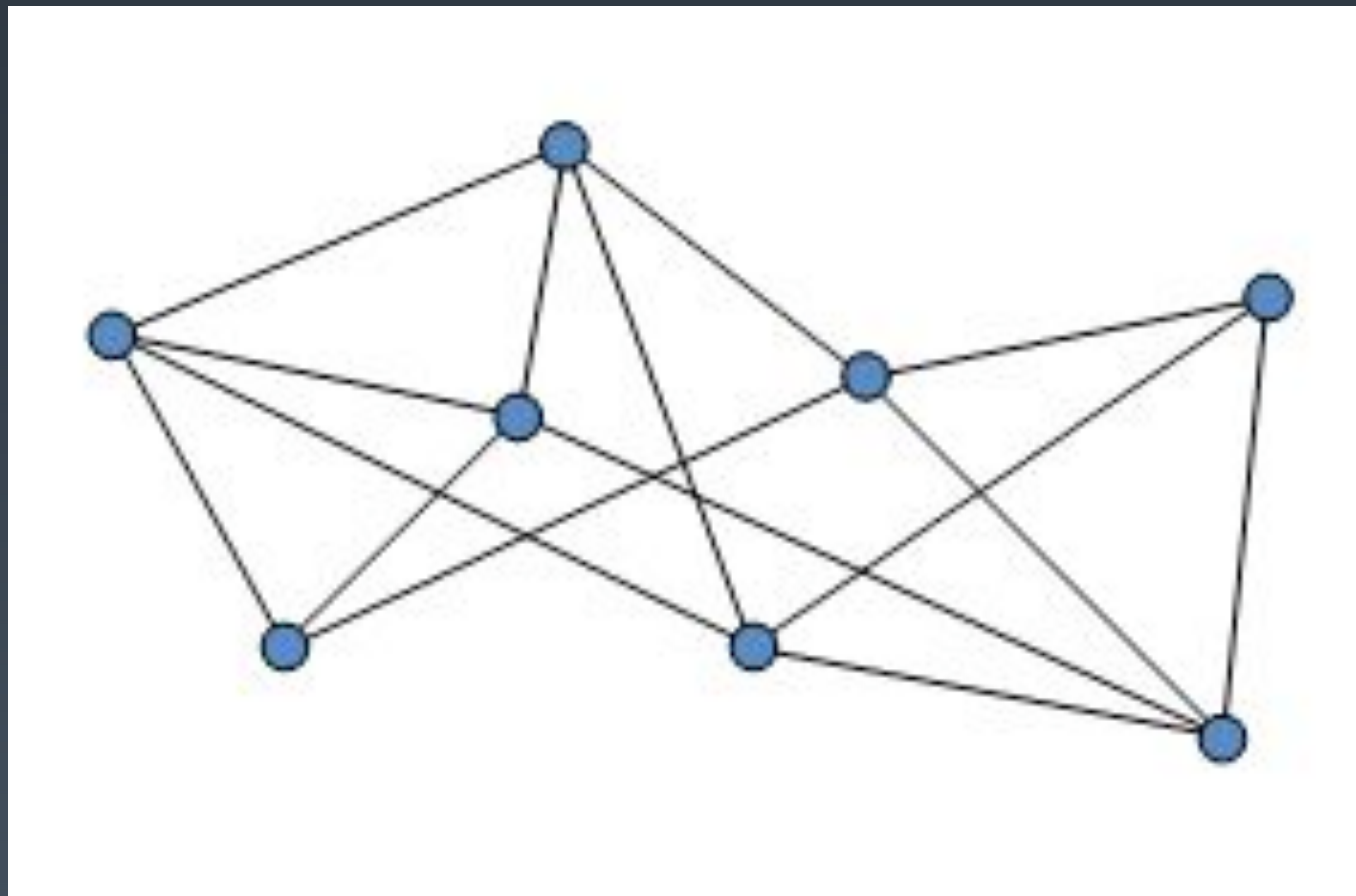
1. 数组的长度

2. 递归的深度（特殊说明）

[https://leetcode-cn.com/problems/climbing-stairs/
solution/pa-lou-ti-by-leetcode/](https://leetcode-cn.com/problems/climbing-stairs/solution/pa-lou-ti-by-leetcode/)

图 Graph

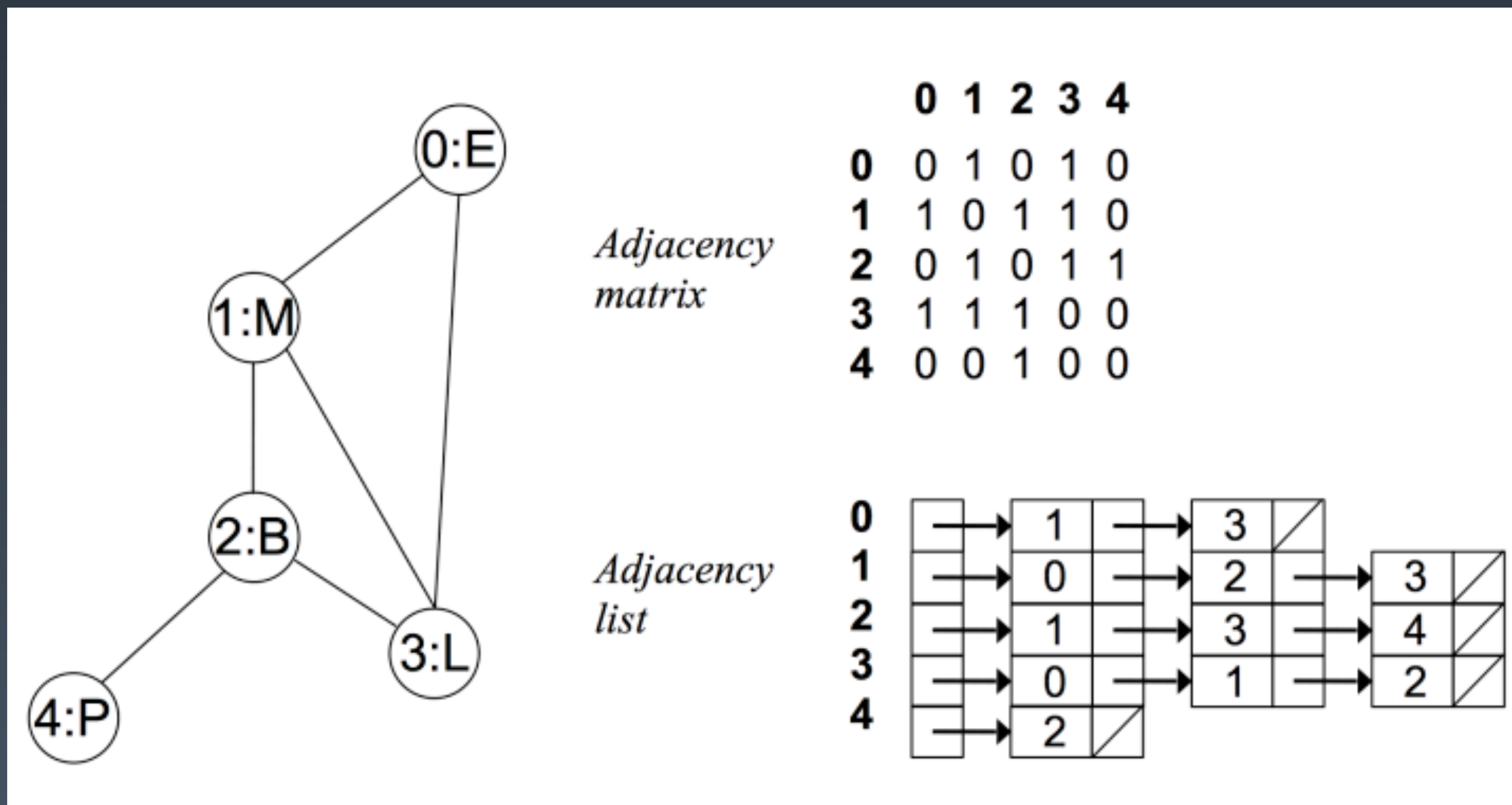
图：点 + 边



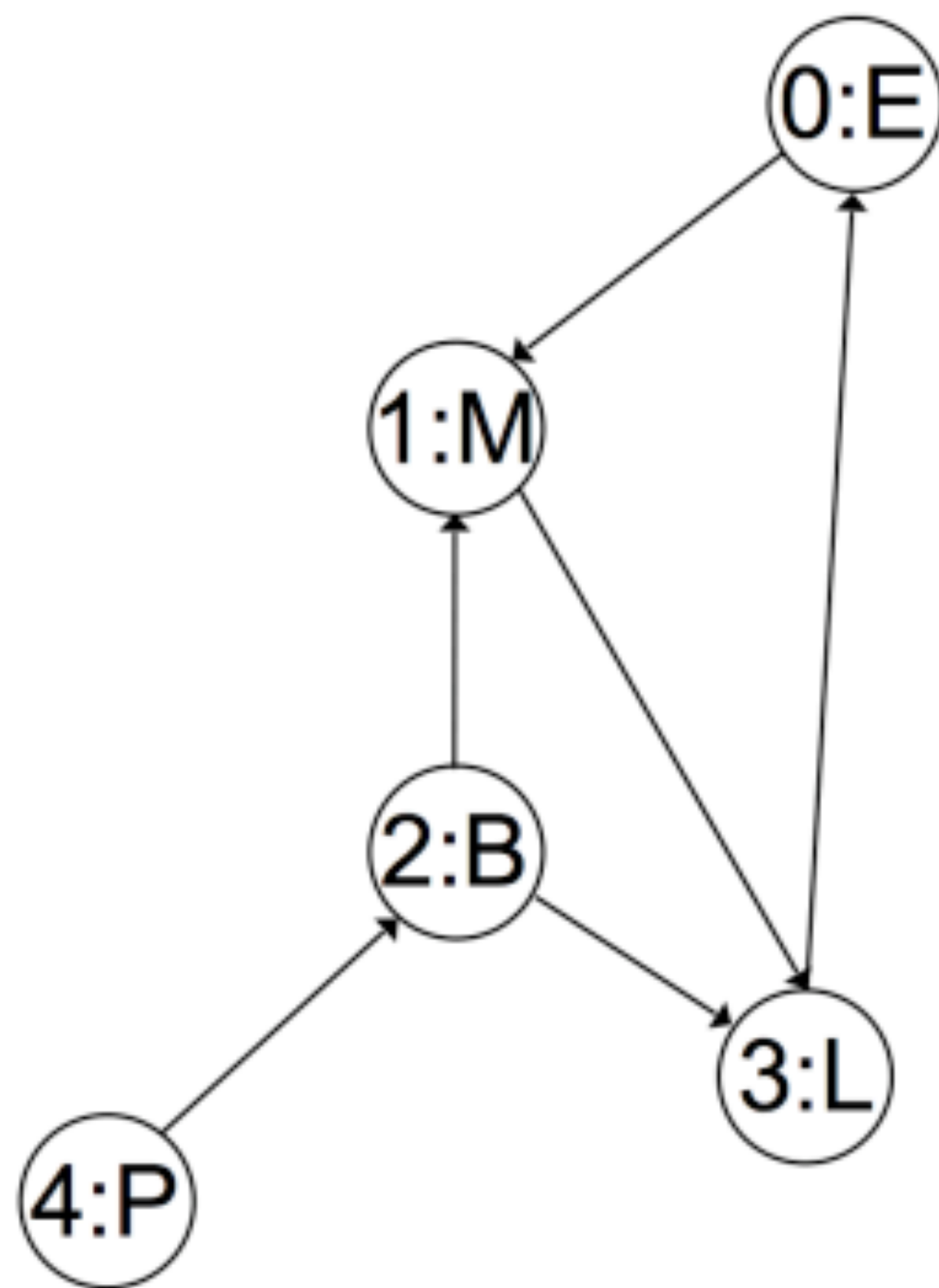
图的定义和分类

- $\text{Graph}(V, E)$: V 顶点、 E 边
- 边:
 - 有向和无向（单行线）
 - 权重（边长）
- 点:
 - 度：入度和出度
 - 连通与否

图：无向无权图



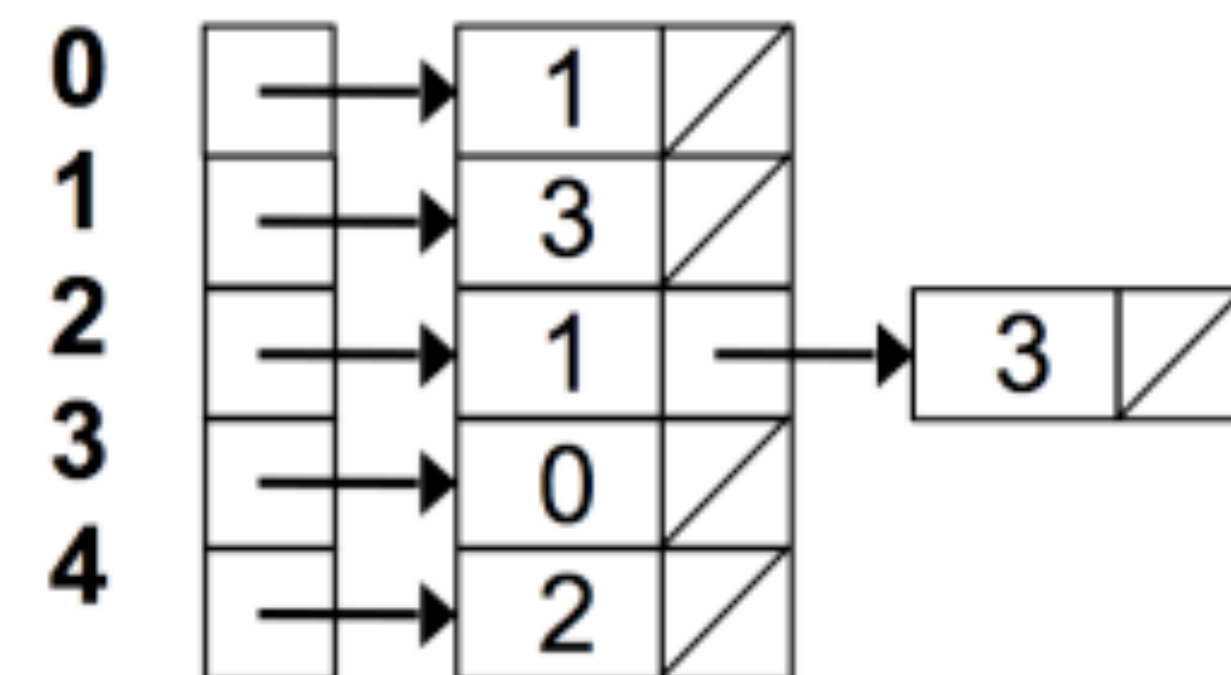
图：有向无权图



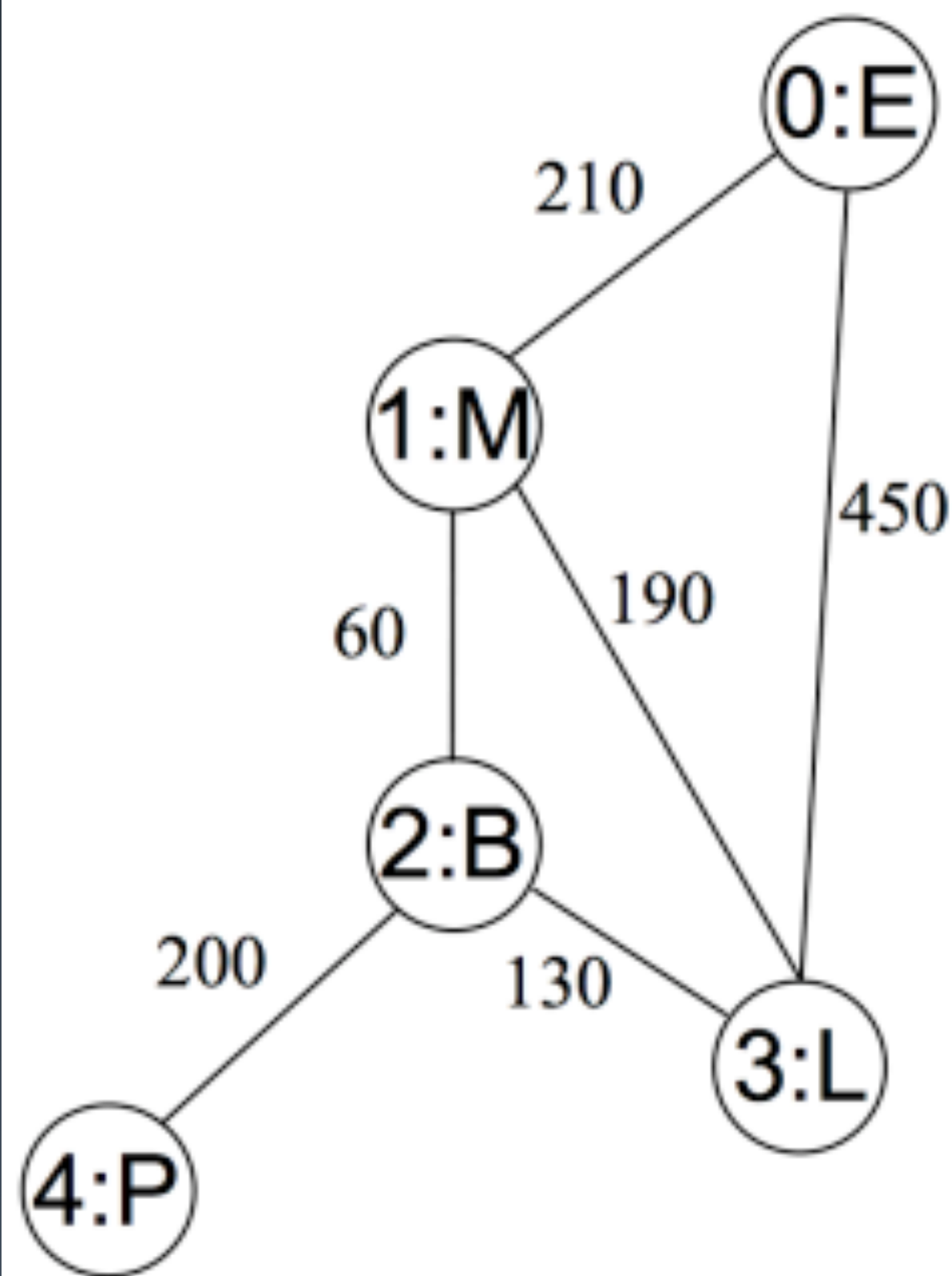
*Adjacency
matrix*

	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*

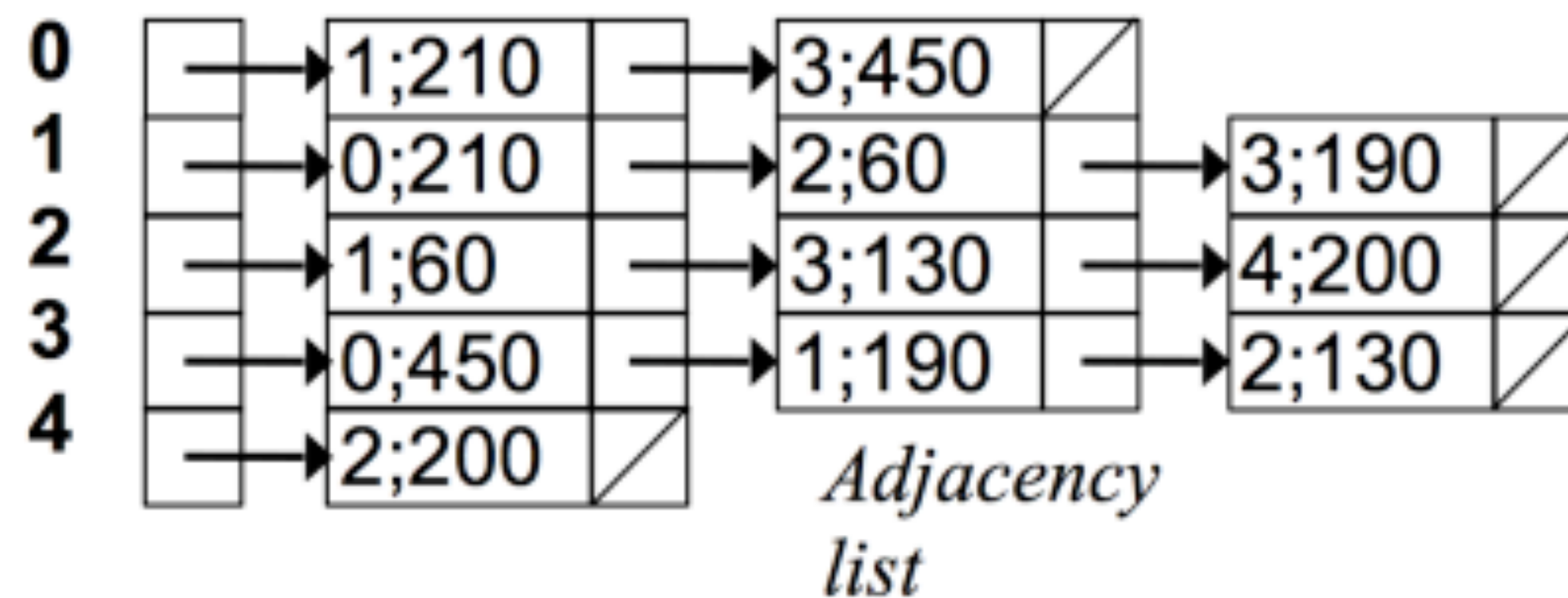


图：无向有权图



	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



DFS 代码 - 递归写法

```
visited = set()
```

```
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])  
  
    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. 最短路径 (Shortest Path) : Dijkstra, Floyd
2. 最小生成树 (Minimum Spanning Tree) : Prim、Kruskal
3. 连通图个数
4. 拓扑排序 (Topological Sorting)

二叉堆

<https://www.cnblogs.com/skywang12345/p/3610187.html#a1>

学习方法上

改变自己的学习习惯（不要死磕）

五毒神掌（重要的是 五！ 过遍数，而不是每次花很长时间）

最大误区：只做一遍！

附带：死磕AC了事，不看高票代码（不看题解）

照着我的话去做

THANKS! |  极客大学