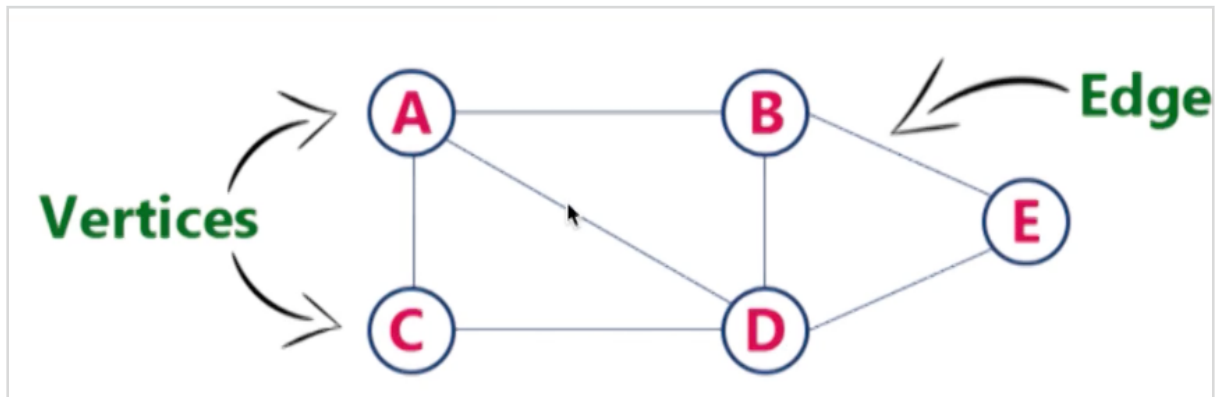


8. 图 (Graph) 的实现和特性

图 Graph

算法面试比较少，少讲，链接题目为面试频率出现较多；工程问题，与业务情况相结合，自己解决

- 定义：有点、有边



- 属性：

- V - vertex 点

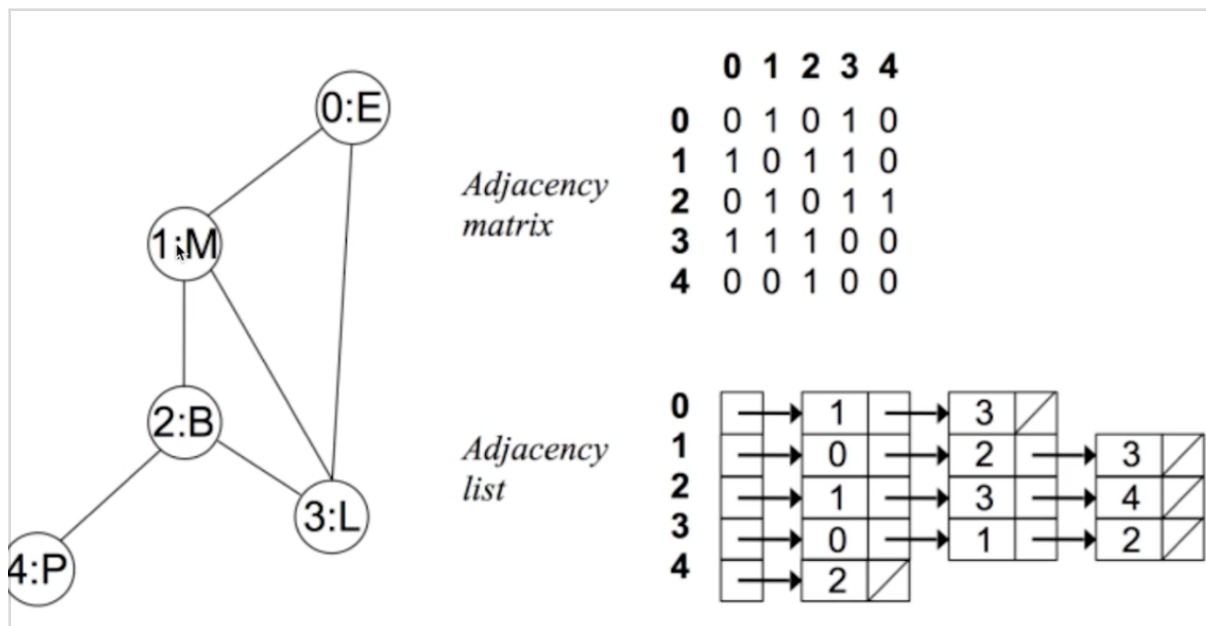
1. 度：入度和出度
2. 点与点之间：连通与否

- E - edge 边

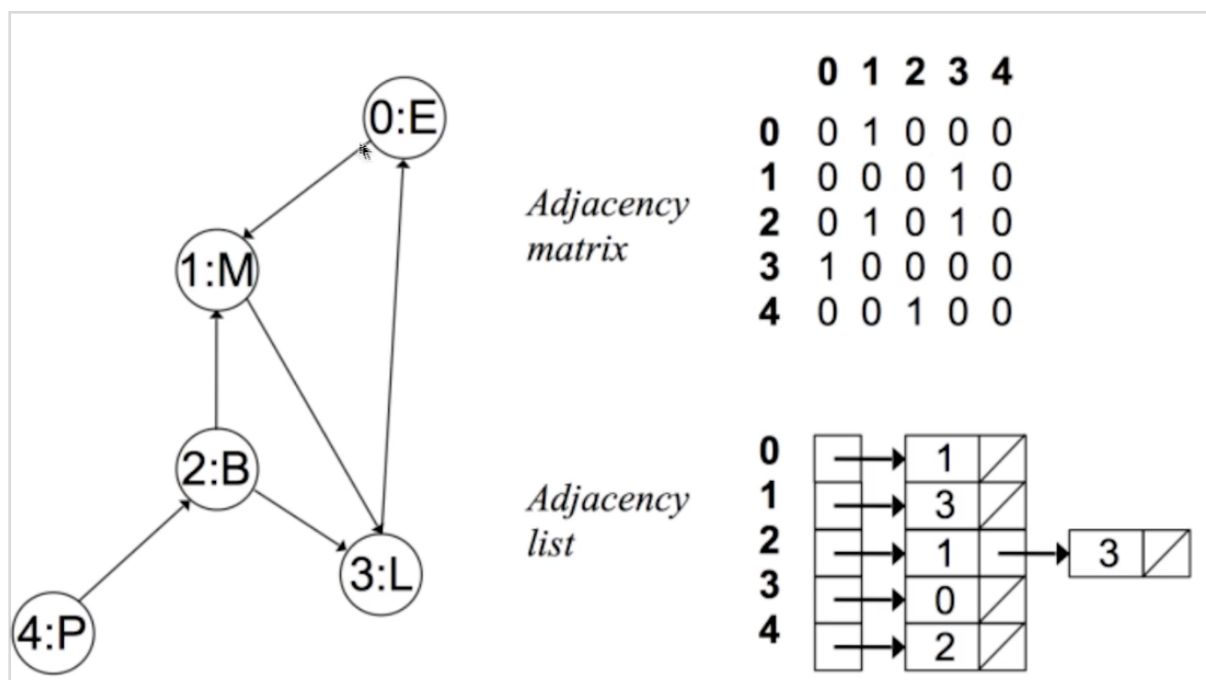
1. 有向：单向
2. 无向：双向
3. 权重：边长

- 分类：

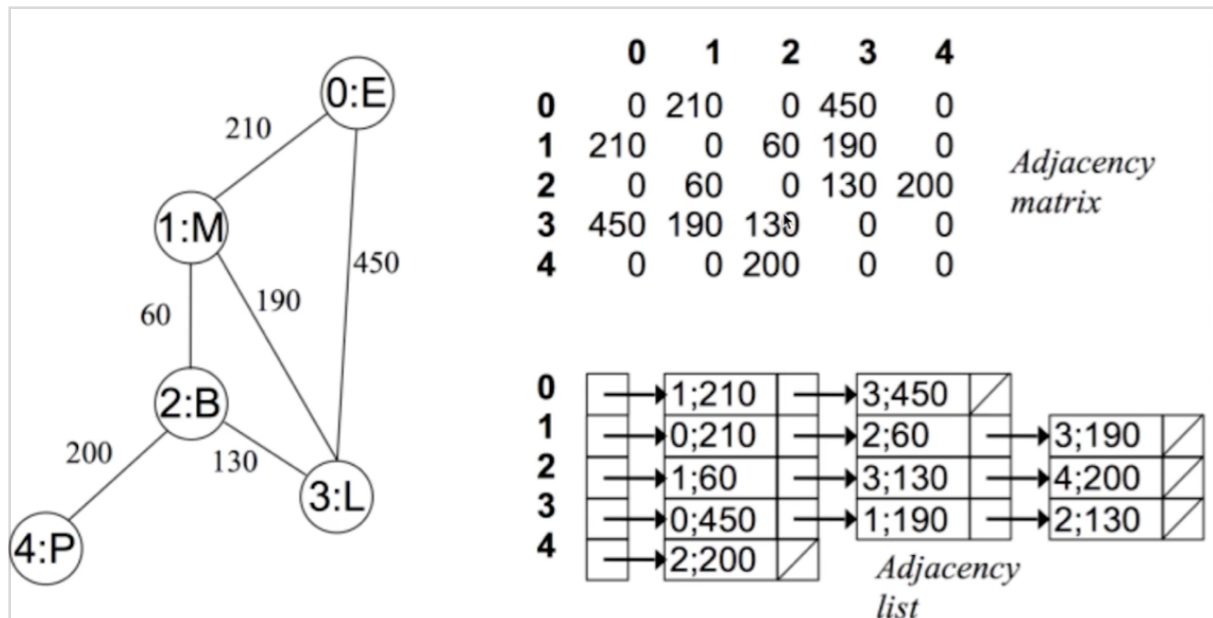
1. 无向无权图



2. 有向无权图



3. 无向有权图



4. 有向有权图

自己画一下

- 基于图相关的算法

树在做 DFS 和 BFS 时，没有环路，顶点不会重复，而图有环路，所以会重复，用 `visited = set()` 来保证顶点不重复。

- DFS 代码 - 递归写法（具体讲解见后面）

```
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
 2. 拓扑排序
 3. 最短路径
 4. 最小生成树