

Design and Analysis of Algorithms

Part IV: Graph Algorithms

Lecture 21: Review of Breadth-First Search

童咏昕

**北京航空航天大学
计算机学院**



- 在算法课程第四部分“图算法”主题中，我们将主要聚焦于如下经典问题：

- Basic Concepts in Graph Algorithms (图算法的基本概念)
- Breadth-First Search (BFS, 广度优先搜索)
- Depth-First Search (DFS, 深度优先搜索)
- Cycle Detection (环路检测)
- Topological Sort (拓扑排序)
- Strongly Connected Components (强连通分量)
- Minimum Spanning Trees (最小生成树)
- Single Source Shortest Path (单源最短路径)
- All-Pairs Shortest Paths (所有点对最短路径)
- Bipartite Graph Matching (二分图匹配)
- Maximum/Network Flows (最大流/网络流)



图的搜索

算法思想

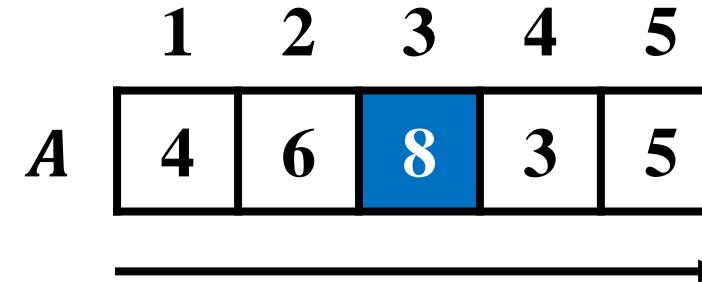
算法实例

算法分析

算法应用

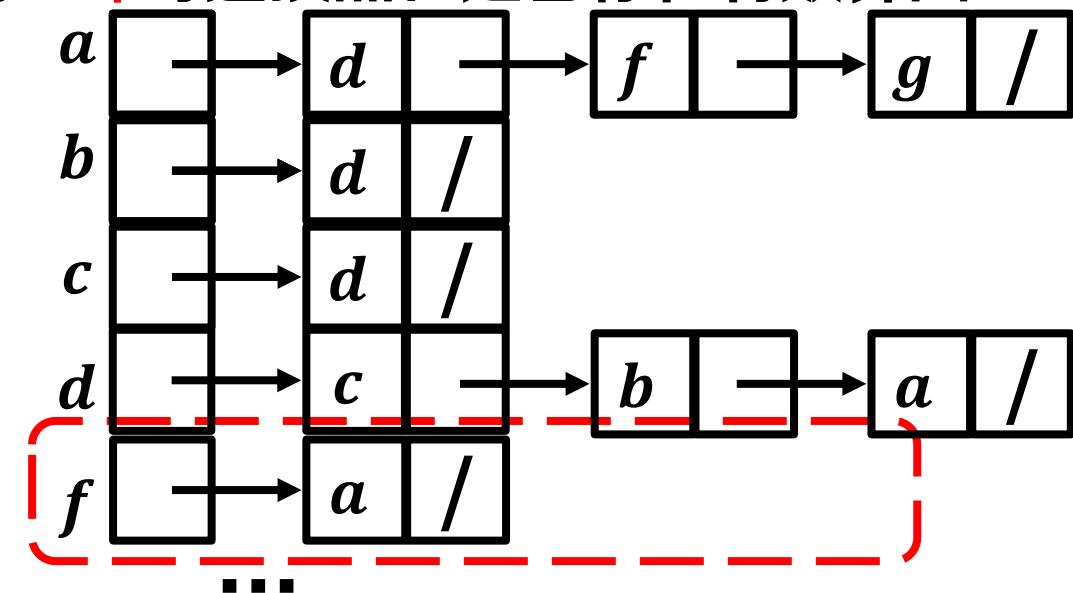
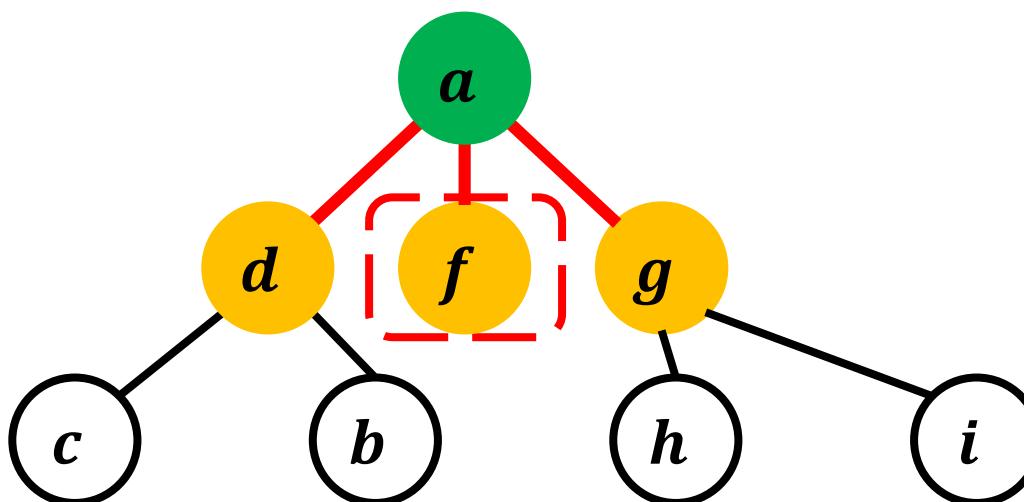
数组结构

- 查询最大值：简单循环搜索所有元素，记录最大值



图结构

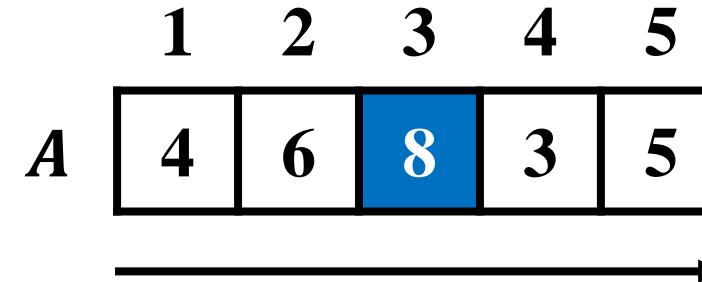
- 查询相邻顶点：简单循环搜索各顶点关联的边
- 查询可达顶点：简单循环搜索，**不能找到全部可达顶点！是否存在有效算法？**



图的搜索

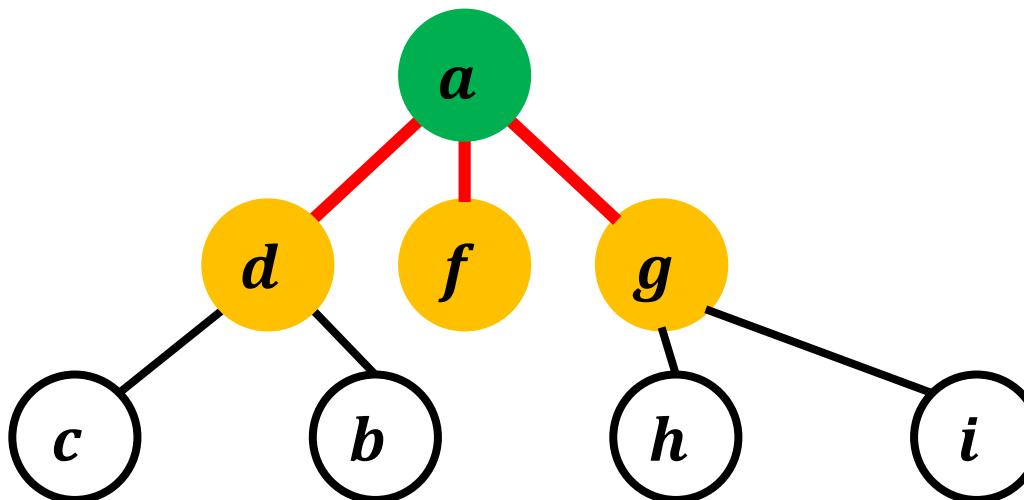
数组结构

- 查询最大值：简单循环搜索所有元素，记录最大值



图结构

- 查询相邻顶点：简单循环搜索各顶点关联的边
- 查询可达顶点：简单循环搜索，不能找到全部可达顶点！是否存在有效算法？



按照什么次序搜索顶点？

广度优先搜索

深度优先搜索



图的搜索

算法思想

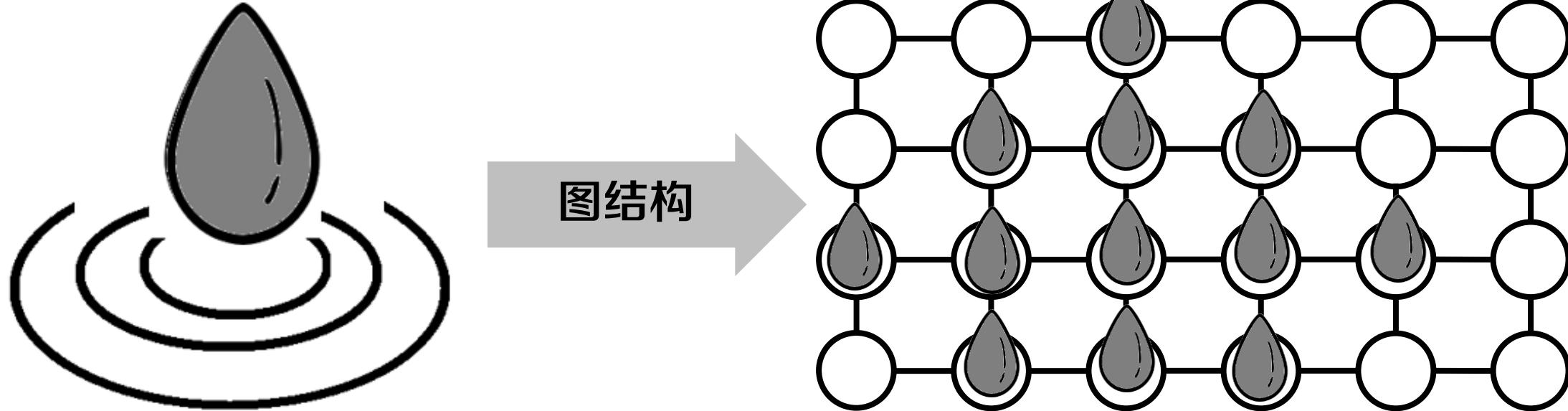
算法实例

算法分析

算法应用

算法思想：现象启发

- 水滴落入水面所激起的涟漪会向相邻区域逐渐扩散

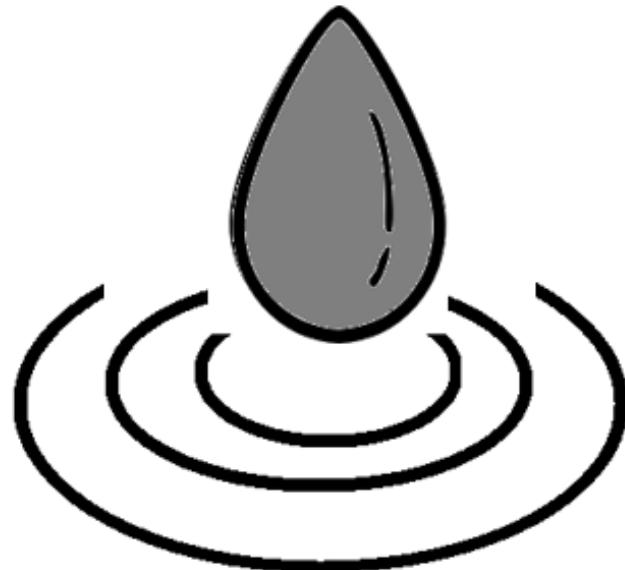


这种依次扩散的现象，蕴含了搜索图结构的一种顺序

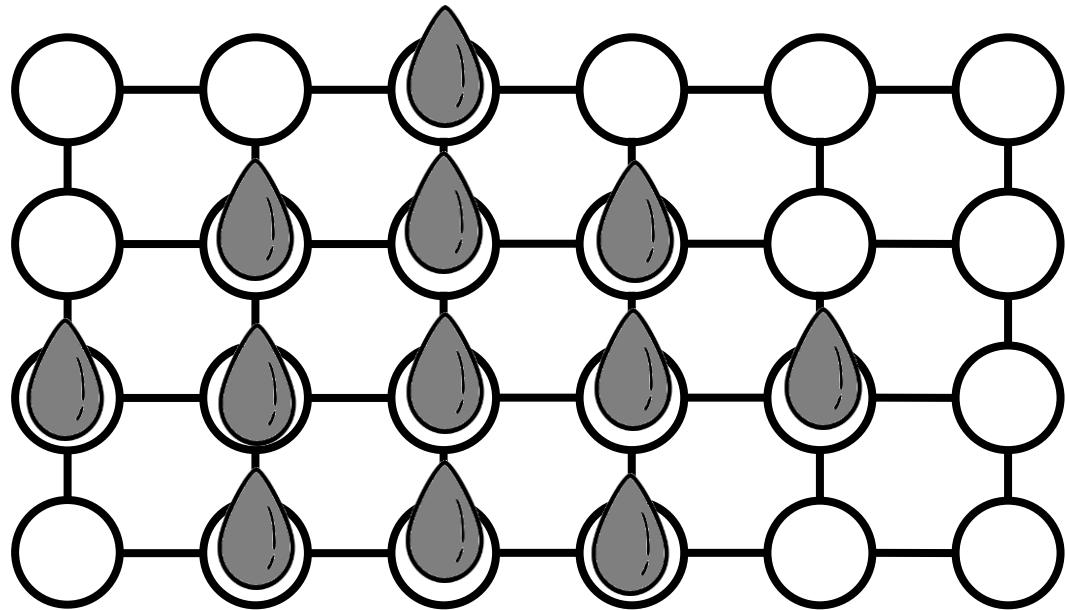
算法思想：现象启发



- 水滴落入水面所激起的涟漪会向相邻区域逐渐扩散



图结构



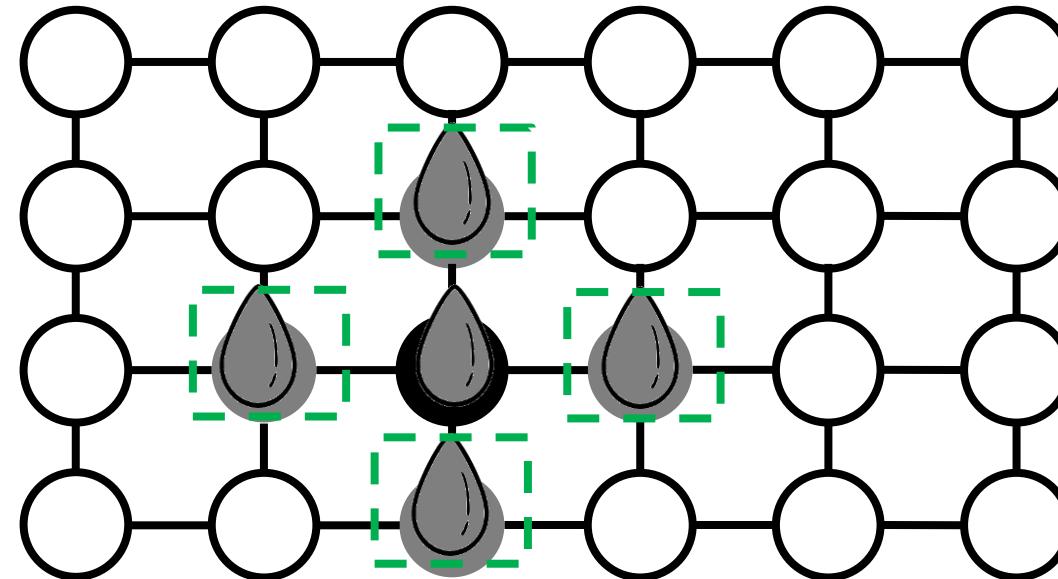
处理某顶点时，一次性发现所有其相邻顶点

算法思想：广度优先搜索



- 核心思想
 - 处理某顶点时，一次性发现其所有相邻顶点

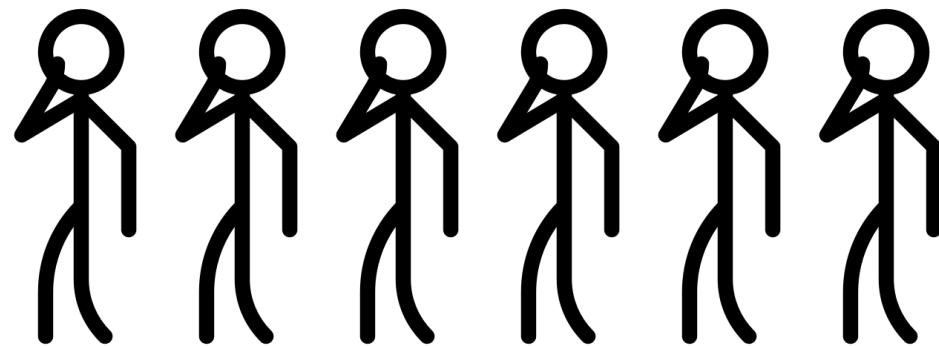
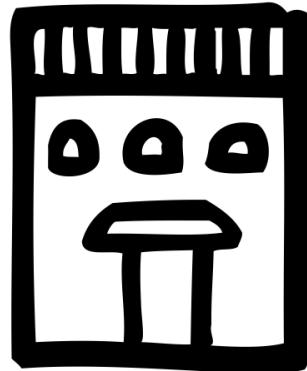
扩散中多处同时进行，一次只能处理单个顶点怎么办？





● 队列

- 先来先服务：队尾加入，队首离开
 - 加入队列， $Q.\text{Enqueue}()$
 - 离开队列， $Q.\text{Dequeue}()$

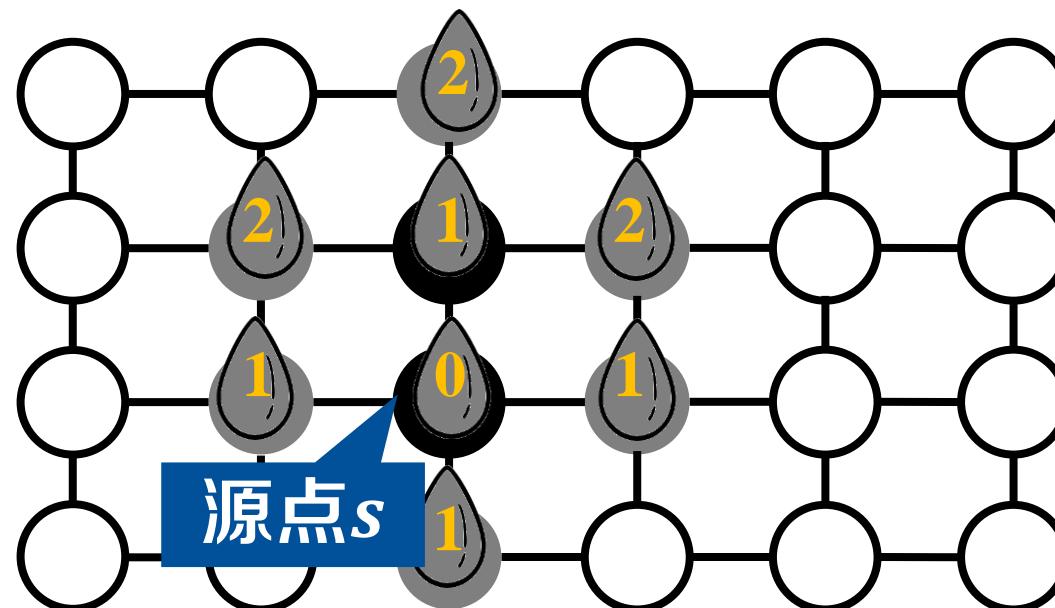


队列 Q

算法思想：广度优先搜索



- 辅助数组
 - $color$ 表示顶点状态
 - **White**: 白色顶点 u 尚未被发现，发现后直接入队
 - **Black**: 黑色顶点 u 已被处理，无需再次入队
 - **Gray**: 灰色顶点 u 已加入队列，无需再次入队
 - $pred$: 顶点 u 由 $pred[u]$ 发现
 - $dist$: 顶点 u 距离源点 s 的距离





图的搜索

算法思想

算法实例

算法分析

算法应用

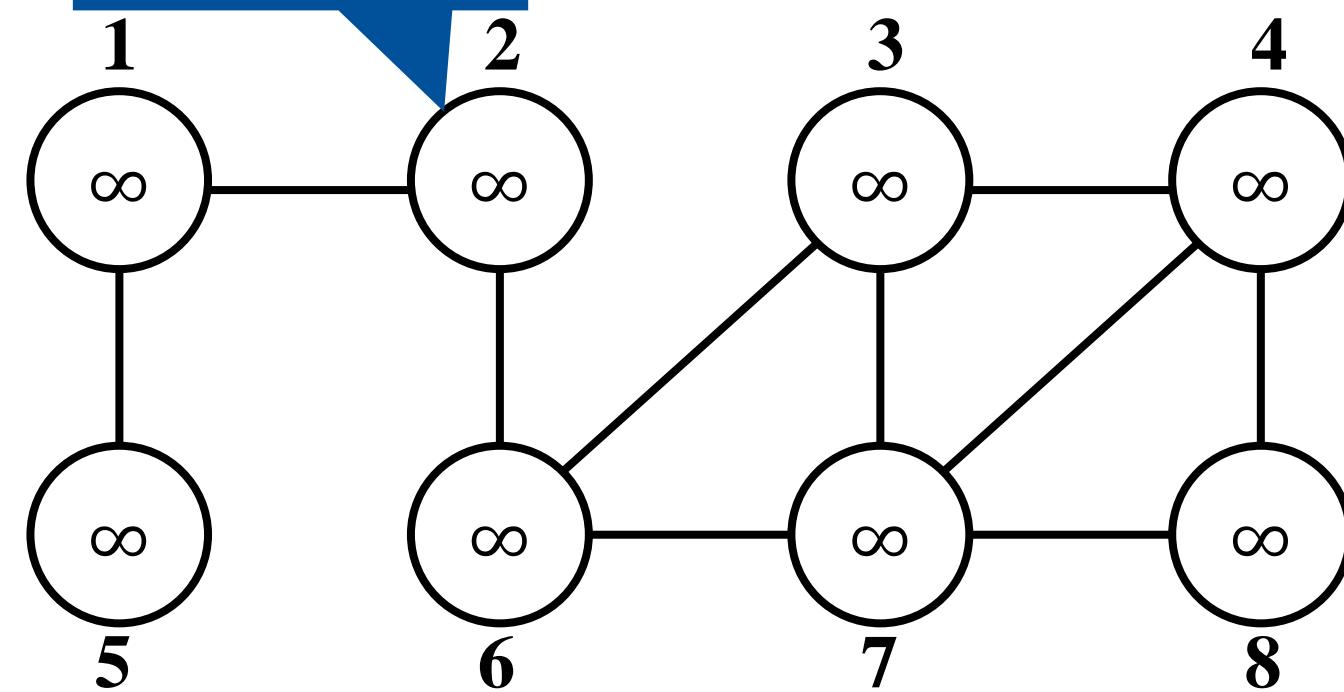
算法实例



V	1	2	3	4	5	6	7	8
$color$	W	W	W	W	W	W	W	W
$pred$	N	N	N	N	N	N	N	N
$dist$	∞							

待处理队列

搜索源点

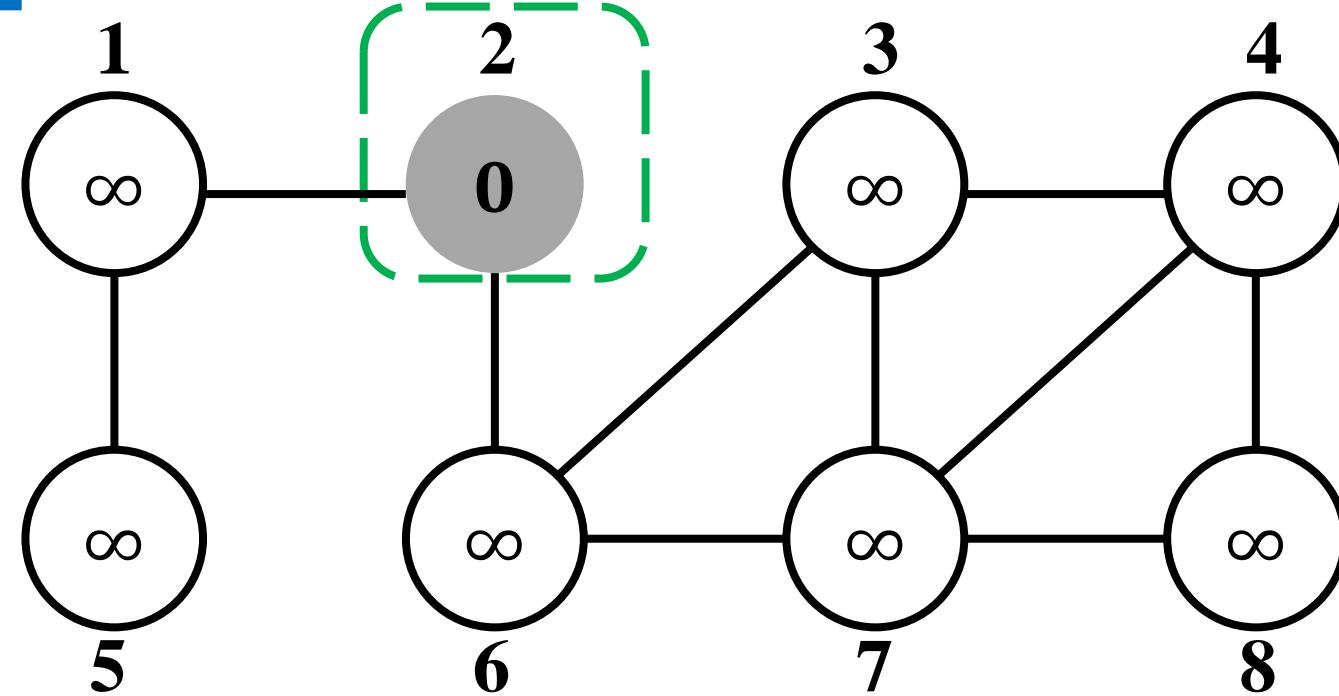


算法实例

V	1	2	3	4	5	6	7	8
$color$	W	G	W	W	W	W	W	W
$pred$	N	N	N	N	N	N	N	N
$dist$	∞	0	∞	∞	∞	∞	∞	∞

待处理队列	2
-------	---

源点2



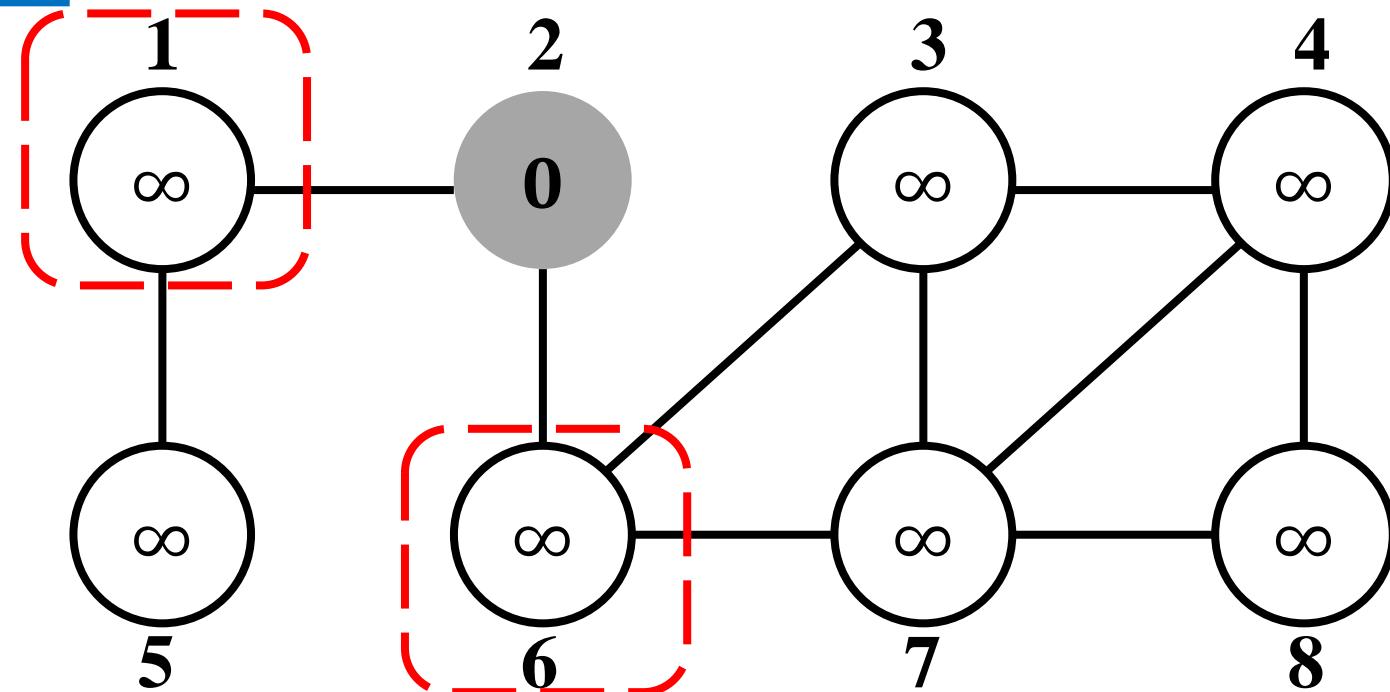
算法实例



V	1	2	3	4	5	6	7	8
$color$	W	G	W	W	W	W	W	W
$pred$	N	N	N	N	N	N	N	N
$dist$	∞	0	∞	∞	∞	∞	∞	∞

待处理队列	2
-------	---

源点2

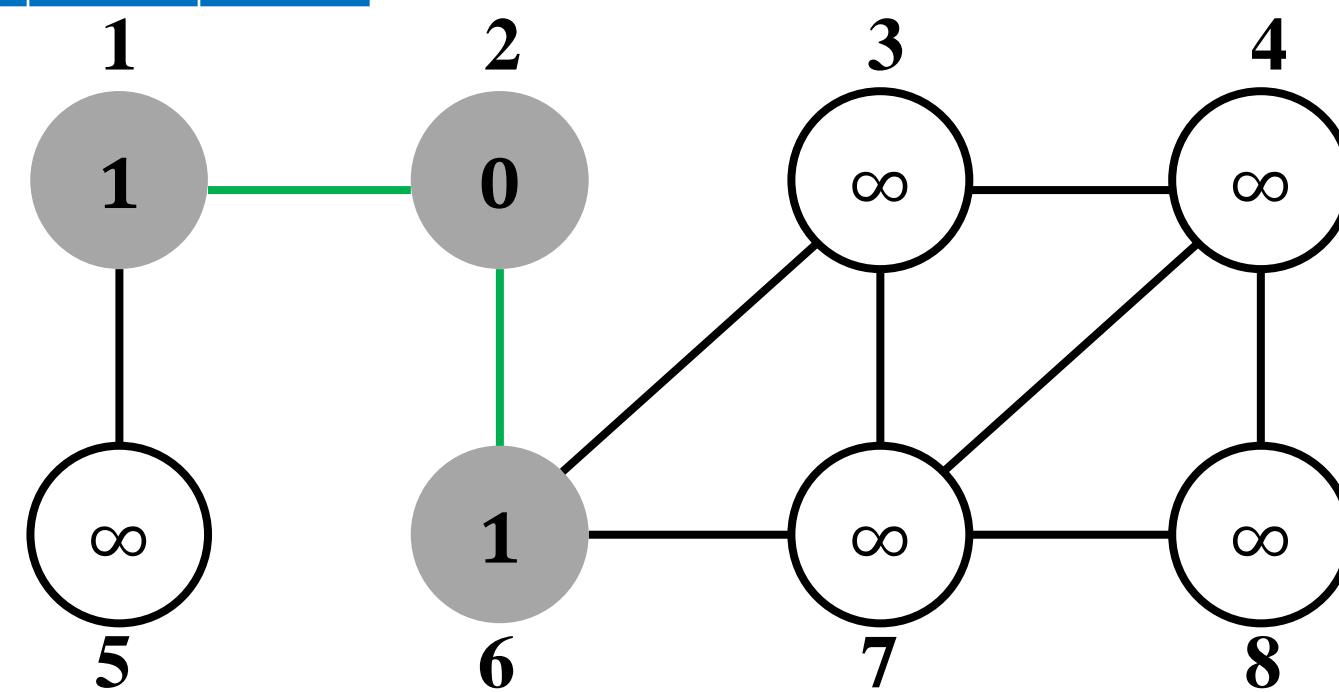


算法实例



V	1	2	3	4	5	6	7	8
$color$	G	G	W	W	W	G	W	W
$pred$	2	N	N	N	N	2	N	N
$dist$	1	0	∞	∞	∞	1	∞	∞
待处理队列	2	1	6					

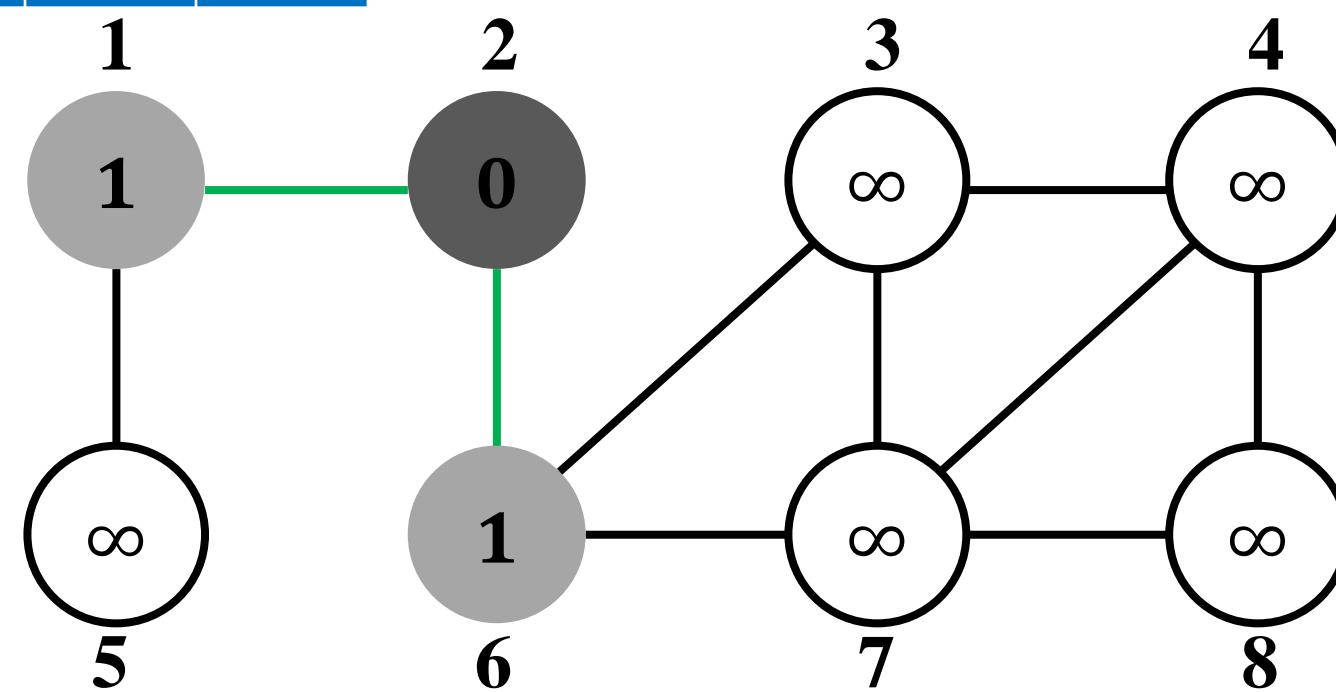
源点2



算法实例

V	1	2	3	4	5	6	7	8
$color$	G	B	W	W	W	G	W	W
$pred$	2	N	N	N	N	2	N	N
$dist$	1	0	∞	∞	∞	1	∞	∞
待处理队列	2 1 6							

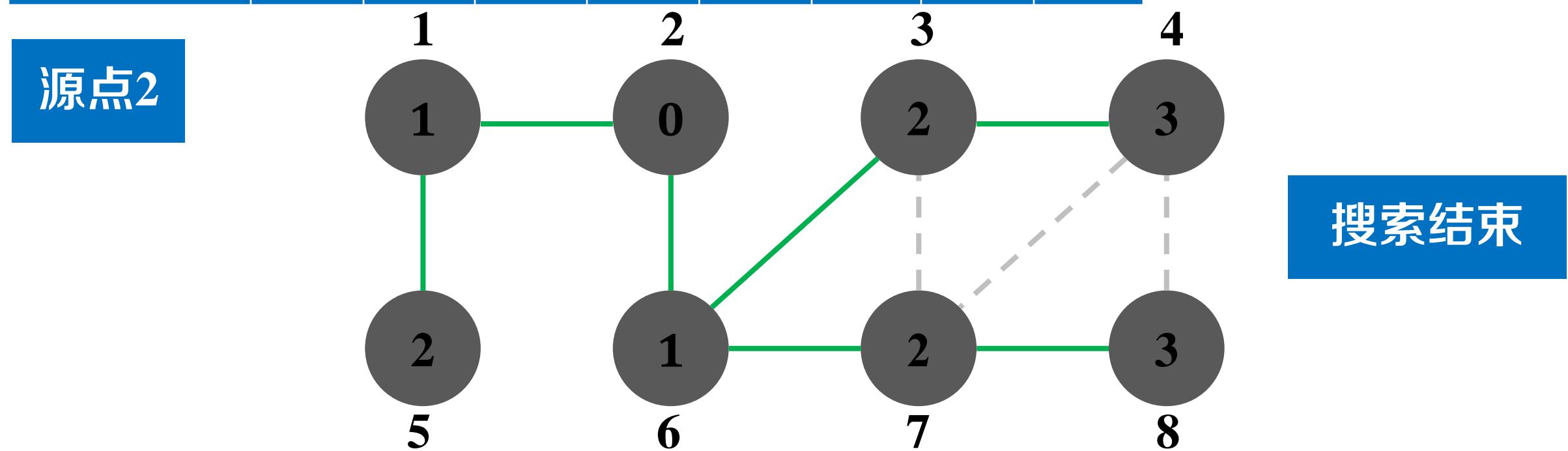
源点2



算法实例



V	1	2	3	4	5	6	7	8
$color$	B	B	B	B	B	B	B	B
$pred$	2	N	6	3	1	2	6	7
$dist$	1	0	2	3	2	1	2	3
待处理队列	2	1	6	5	3	7	4	8

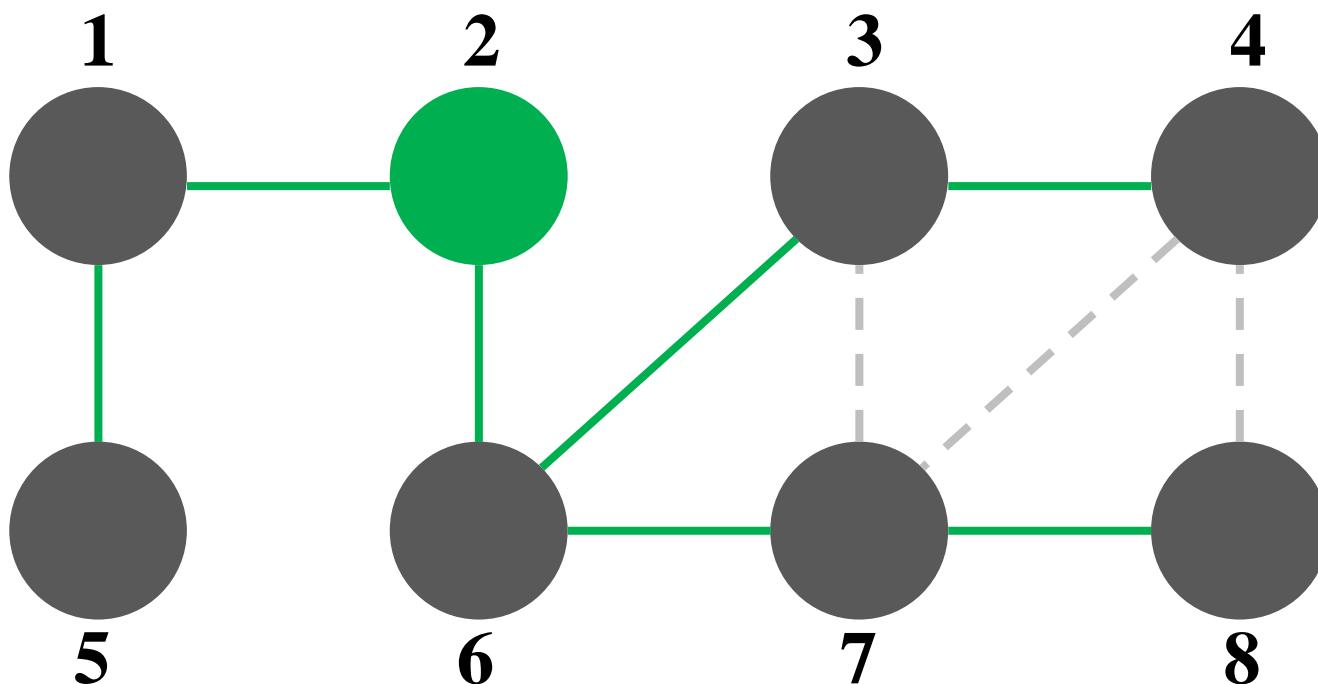


广度优先树

V	1	2	3	4	5	6	7	8
$pred$	2	N	6	3	1	2	6	7

- 辅助数组 $pred[]$ 储存了一棵树，**广度优先树** $T = < V_T, E_T >$

连通、无环
 $|E_T| = |V_T| - 1$



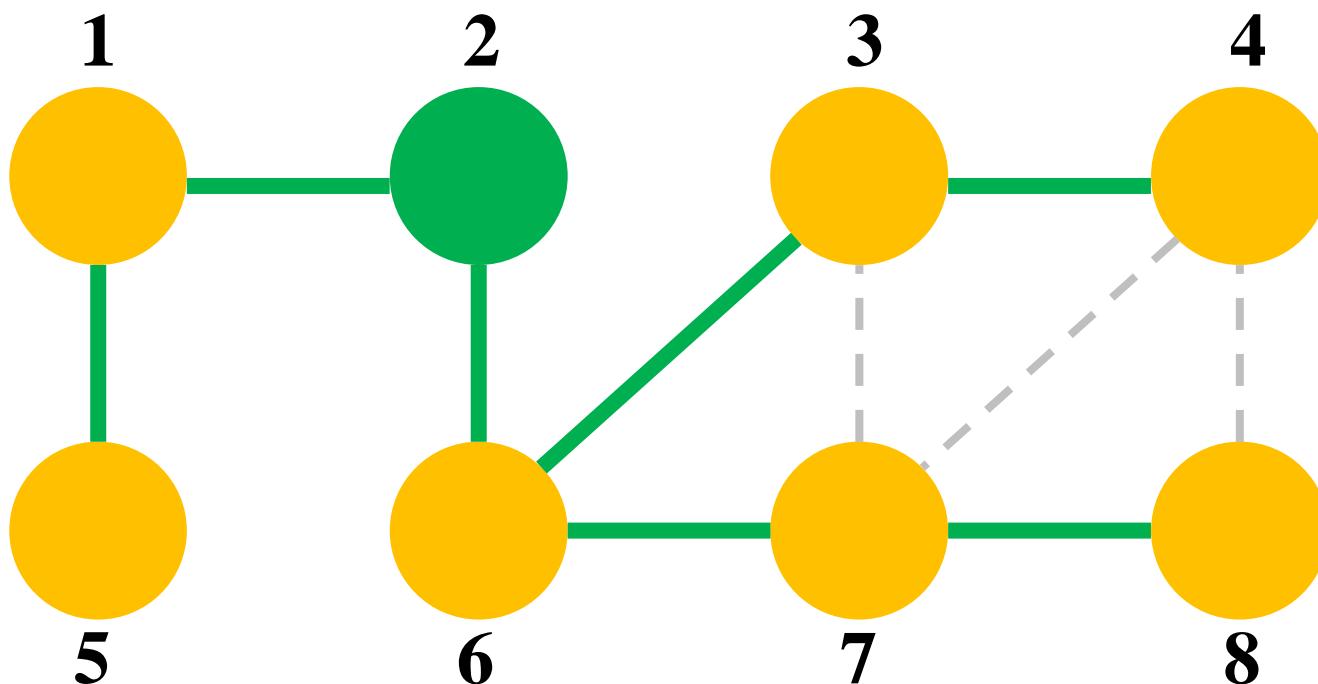
广度优先树

V	1	2	3	4	5	6	7	8
$pred$	2	N	6	3	1	2	6	7

- 辅助数组 $pred[]$ 储存了一棵树， 广度优先树 $T = < V_T, E_T >$
- V_T 是源点 s 可达的顶点集合， $E_T = \{(pred[u], u) : u \in V_T\}$

连通、无环

$$|E_T| = |V_T| - 1$$





图的搜索

算法思想

算法实例

算法分析

算法应用



伪代码

- **BFS(G, s)**

输入: 图 G , 源点 s

输出: 前驱数组 $pred[]$, 距离数组 $dist[]$

新建一维数组 $color[1..|V|]$, $pred[1..|V|]$, $dist[1..|V|]$

新建空队列 Q

//初始化

for $u \in V$ do

$color[u] \leftarrow WHITE$

$pred[u] \leftarrow NULL$

$dist[u] \leftarrow \infty$

end

$color[s] \leftarrow GRAY$

$dist[s] \leftarrow 0$

$Q.Enqueue(s)$



伪代码

- **BFS(G, s)**

```
//广度优先搜索
while 等待队列Q非空 do
     $u \leftarrow Q.Dequeue()$ 
    for  $v \in G.Adj[u]$  do
        if  $color[v] = WHITE$  then
             $color[v] \leftarrow GRAY$ 
             $dist[v] \leftarrow dist[u] + 1$ 
             $pred[v] \leftarrow u$ 
             $Q.Enqueue(v)$ 
        end
    end
     $color[u] \leftarrow BLACK$ 
end
```



复杂度分析

- 对于每个顶点 u , 搜索相邻顶点消耗时间 $T_u = O(1 + \deg(u))$
- 总运行时间:

$$\begin{aligned} T &\leq \sum_{u \in V} T_u \\ &= \sum_{u \in V} O(1 + \deg(u)) \\ &= \sum_{u \in V} O(1) + \sum_{u \in V} O(\deg(u)) \\ &= O(|V| + |E|) \end{aligned}$$

在渐近记号中, 约定符号 V 代表 $|V|$, 符号 E 代表 $|E|$



复杂度分析

- 对于每个顶点 u , 搜索相邻顶点消耗时间 $T_u = O(1 + \deg(u))$
- 总运行时间:

$$\begin{aligned} T &\leq \sum_{u \in V} T_u \\ &= \sum_{u \in V} O(1 + \deg(u)) \\ &= \sum_{u \in V} O(1) \quad \text{简化表示} \\ &= O(V + E) \end{aligned}$$

在渐近记号中, 约定符号 V 代表 $|V|$, 符号 E 代表 $|E|$



图的搜索

算法思想

算法实例

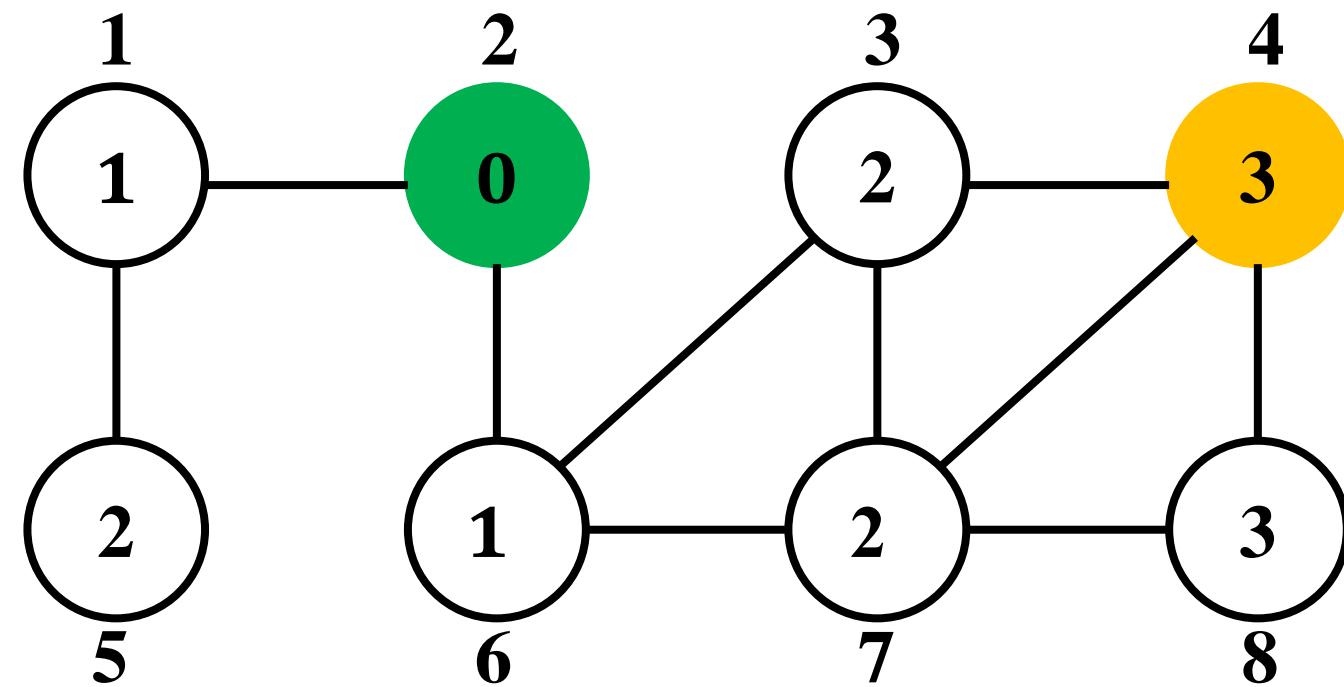
算法分析

算法应用

相关应用：最短路径

V	1	2	3	4	5	6	7	8
$pred[]$	2	N	6	3	1	2	6	7
$dist[]$	1	0	2	3	2	1	2	3

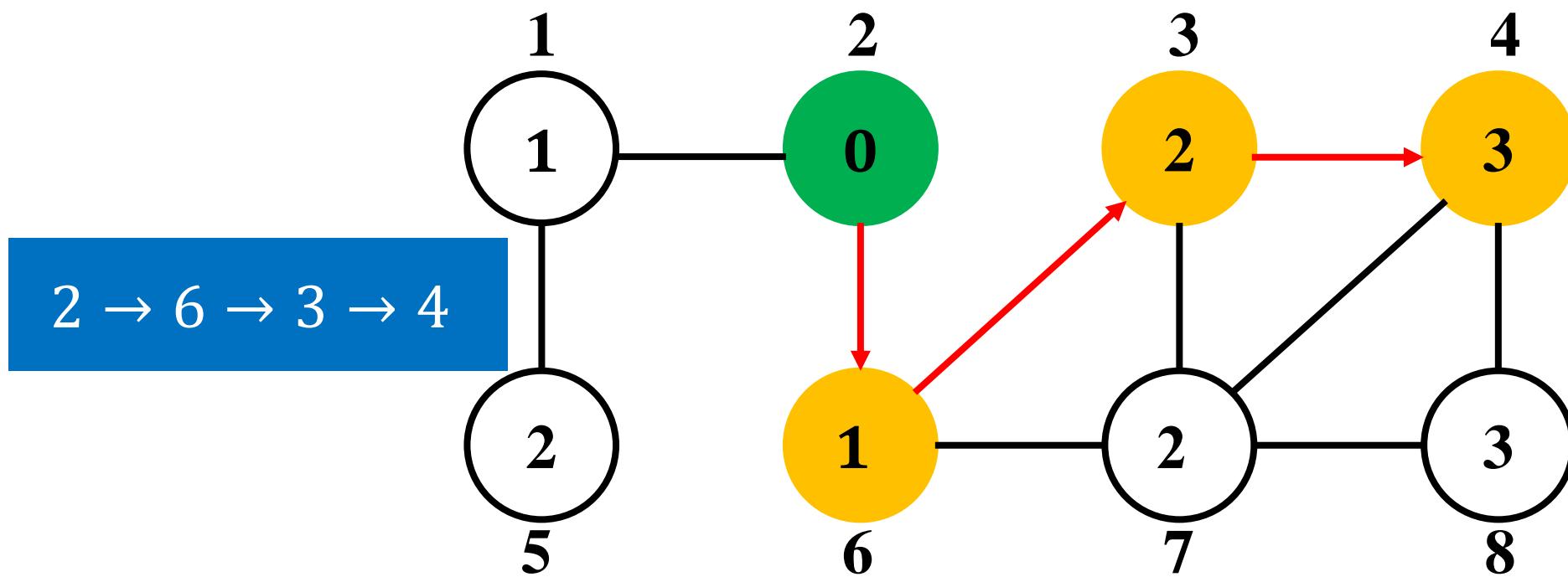
无权图 G 中，源点2到顶点4的最短路径



相关应用：最短路径

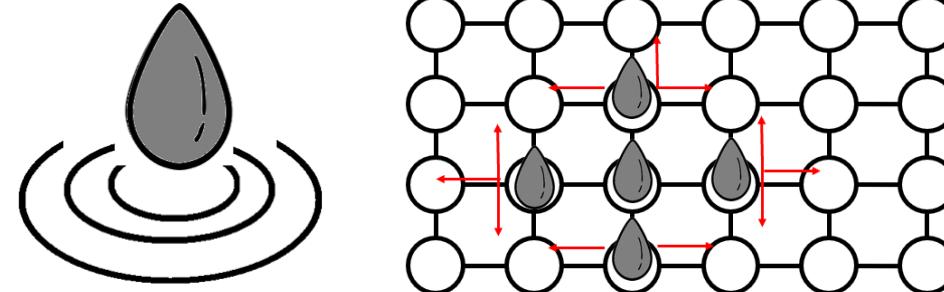
V	1	2	3	4	5	6	7	8
$pred[]$	2	N	6	3	1	2	6	7
$dist[]$	1	0	2	3	2	1	2	3

无权图 G 中，源点2到顶点4的最短路径

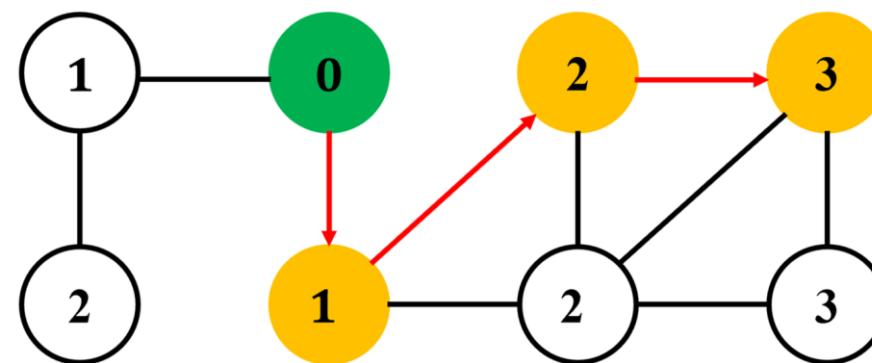


- 广度优先搜索

- 算法核心思想：逐层扩散



- 算法运行时间： $O(|V| + |E|)$ ，简记为 $O(V + E)$
- 算法相关应用：计算最短路径





謝謝

