

6. 图

广度优先搜索
推广

邓俊辉

deng@tsinghua.edu.cn

连通分量 + 可达分量

❖ 问题

给定无向图，找出其中任一顶点s所在的连通分量

给定有向图，找出源自其中任一顶点s的可达分量

❖ 算法

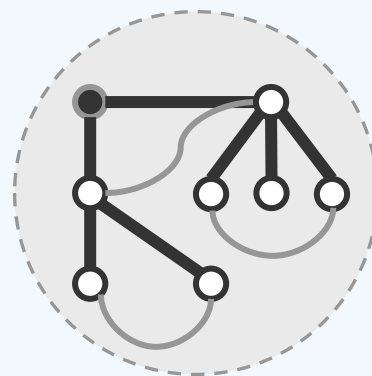
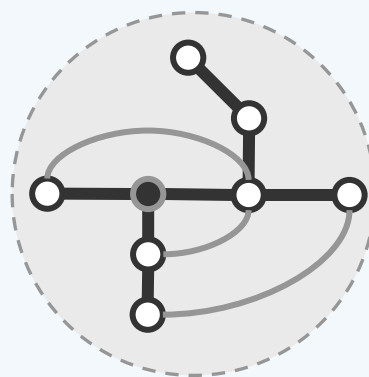
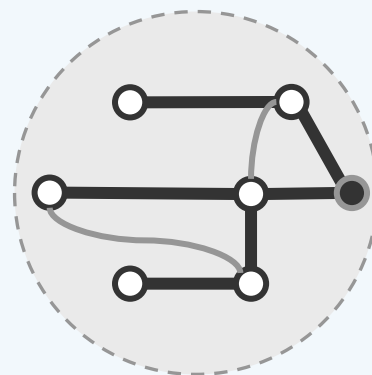
从s出发做BFS

输出所有被发现的顶点

队列为空后立即终止，无需考虑其它顶点

❖ 若图中包含多个连通/可达分量

又该如何保证对全图的遍历呢？



Graph::bfs()

❖ template <typename Tv, typename Te> //顶点类型、边类型

void Graph<Tv, Te>::bfs(int s) { //s为起始顶点

reset(); int clock = 0; int v = s; //初始化 $\Theta(n + e)$

do //逐一检查所有顶点，一旦遇到尚未发现的顶点

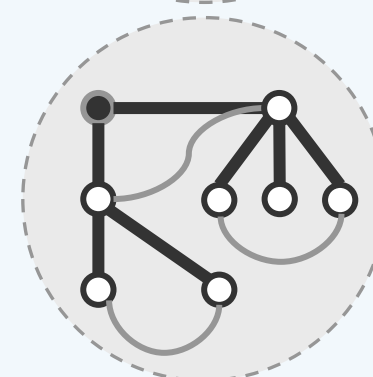
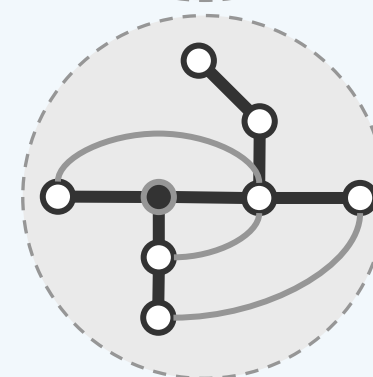
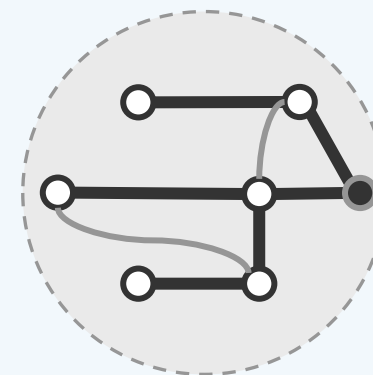
if (UNDISCOVERED == status(v)) //累计 $\Theta(n)$

BFS(v, clock); //即从该顶点出发启动一次BFS

while (s != (v = (++v % n))); //按序号访问，不漏不重

} //无论共有多少连通/可达分量...

❖ bfs()均可遍历它们，而且自身累计仅需线性时间...



复杂度

❖ 考查无向图...

❖ bfs()的初始化 (reset()) $O(n + e)$

❖ BFS()的迭代 $O(n + 2e)$

外循环 (while (!Q.empty())) , 每个顶点只进入1次 , 累计n次 $O(n)$

内循环 (枚举v的每一邻居) , 每个邻居至多进入1次 , 累计deg(v)次

采用邻接矩阵 $O(n)$

采用邻接表 $O(1 + \deg(v))$

总共 = $O(\sum_v (1 + \deg(v))) = O(n + 2e)$

❖ 整个算法 : $O(n + e) + O(n + 2e) = O(n + e)$

❖ 有向图呢 ? 亦是如此 !