06-F1

6.图

拓扑排序

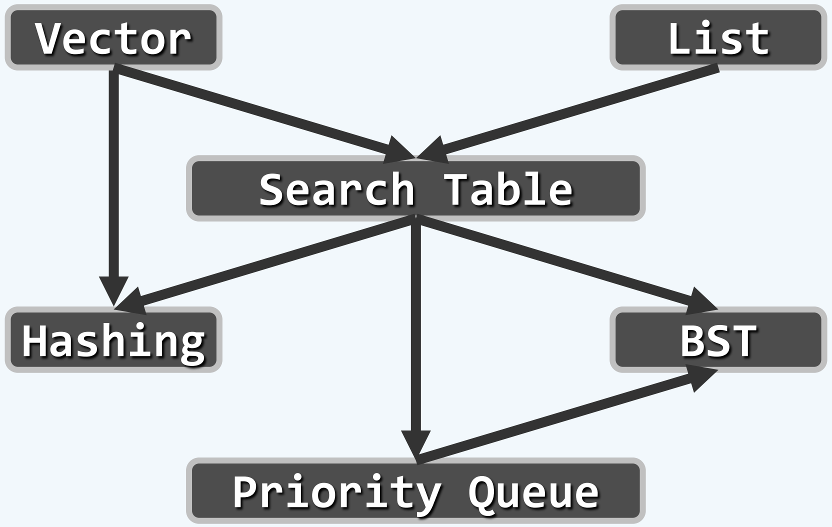
零入度算法

邓俊辉

deng@tsinghua.edu.cn

有向无环图

- Directed Acyclic Graph



- 应用

  类派生和继承关系图中，是否存在循环定义

  操作系统中，相互等待的一组线程可否调度，如何调度

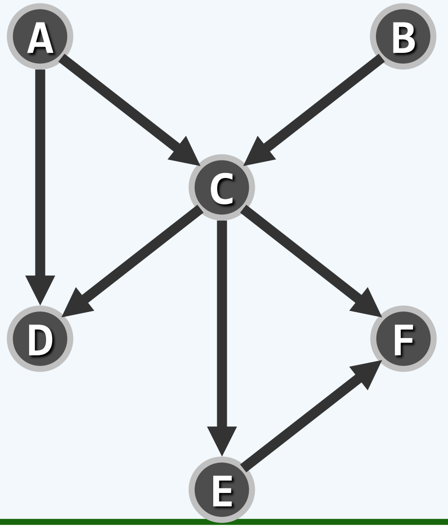
  给定一组相互依赖的课程，是否存在可行的培养方案

  给定一组相互依赖的知识点，是否存在可行的教学进度方案

  项目工程图中，是否存在可串行施工的方案

  email系统中，是否存在自动转发或回复的回路

  ...



拓扑排序

❖ 任给有向图G，不一定是DAG

❖ 尝试将其中顶点排成一个线性序列

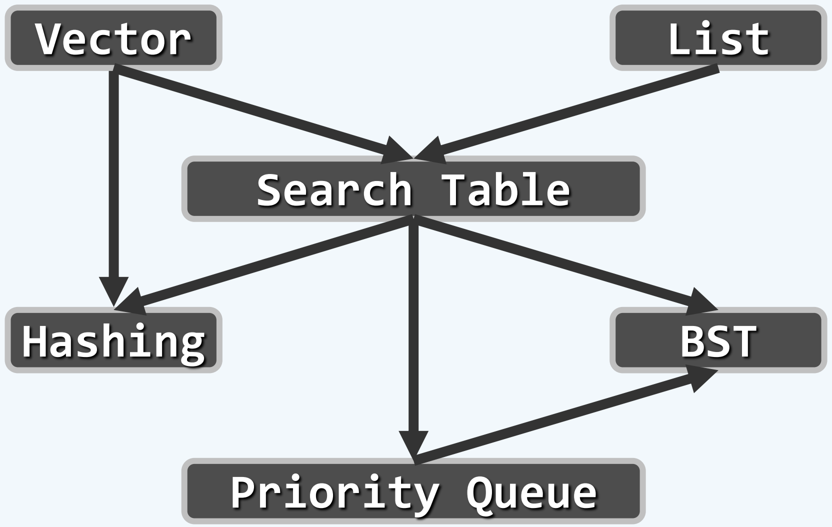
其次序须与原图相容，亦即

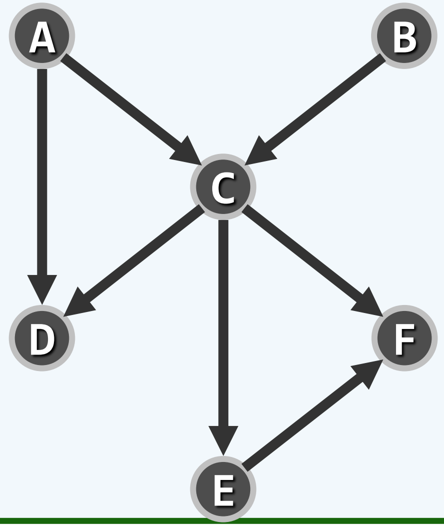
每一顶点都不会通过边指向前驱顶点

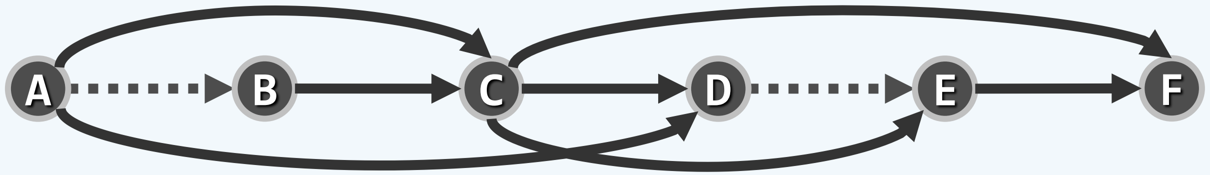
❖ 算法要求

若原图存在回路（即并非DAG），检查并报告

否则，给出一个相容的线性序列







存在性

- 每个DAG对应于一个 $\boxed{\text{偏序集}}$ ；拓扑排序对应于一个 $\boxed{\text{全序集}}$

  所谓的拓扑排序，即构造一个与指定偏序集 $\boxed{\text{相容}}$ 的全序集

- 可以拓扑排序的有向图，必定无环 //反之

  任何DAG，都存在（至少）一种拓扑排序？是的！ //为什么...

- 有限偏序集必有 $\boxed{\text{极大}}$ / $\boxed{\text{极大}}$ 元素

  任何DAG都存在（至少）一种拓扑排序

- 可归纳证明，并直接导出一个算法...

存在性

1. 任何DAG，必有（至少一个）顶点入度为零 //记作m

2. 若 DAG \ {m} 存在拓扑排序 $S = \langle u_{k1}, \ldots, u_{k(n-1)} \rangle$ //subtraction

   则 $S' = \langle m, u_{k1}, \ldots, u_{k(n-1)} \rangle$ 即为DAG的拓扑排序 //DAG子图亦为DAG

❖ 只要m不唯一，拓扑排序也应不唯一 //反之呢？

算法A：顺序输出零入度顶点

```
将所有入度为零的顶点存入栈S，取空队列Q //𝒪(n)
while ( ! S.empty() ) { //𝒪(n)
   Q.enqueue( v = S.pop() ); //栈顶v转入队列
   for each edge( v, u ) //v的邻接顶点u若入度仅为1
      if ( u.inDegree < 2 ) S.push( u ); //则入栈
   G = G \ { v }; //删除v及其关联边（邻接顶点入度减1）
} //总体𝒪(n + e)
return |G| ? Q : "NOT_A_DAG"; //残留的G空，当且仅当原图可拓扑排序
```