第11周小结

1 生成树和最小生成树

❶ 生成树和最小生成树定义

带权连通图 —极小连通子图→ 生成树 —所有边权值和最小→ 最小生成树

❷ 构建生成树的方法

- 深度优先遍历 ⇨ 深度优先生成树
- 广度优先遍历 ⇨ 广度优先生成树

广度优先生成树高度 ≤ 深度优先生成树高度

若一个具有n个顶点和e条边的无向图是一个森林（$n>e$），则该森林必有（　）棵树。

A.e　　B.n　　C.$n-e$　　D.1

- 设该森林有m棵树，节点个数分别为n_1、n_2、…、n_m
- 总节点数$n = n_1 + n_2 + \cdots + n_m$
- 第i棵树的边数=n_i-1（可以看成自己的生成树）
- 总边数 = $(n_1-1)+(n_2-1)+\cdots+(n_m-1) = n-m = e$，所以$m = n-e$

C

❸ 构建最小生成树的算法

☑ Prim算法

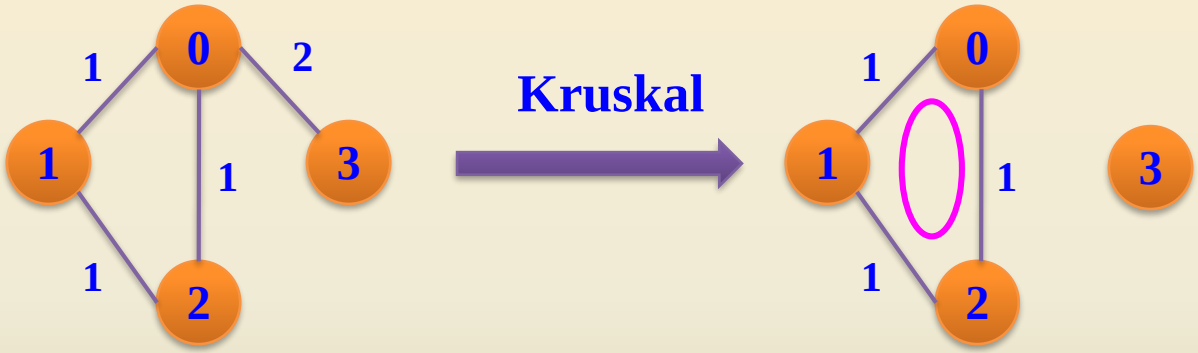
- 起点v
- 所有顶点分为U（$v \in U$）和V-U
- 每次选择这两个集合之间的最小边
- $O(n^2)$

☑ Kruskal算法

- 将边按权值递增排列
- 每次选择权值小并且不构成回路的边
- $O(e\log_2 e)$

一个带权连通图中所有权值最小的边一定会出现在所有的最小生成树中？

不一定！



对某个带权连通图构造最小生成树，以下说法中正确的是（）。

Ⅰ.该图的所有最小生成树的总代价一定是唯一的 √

Ⅱ.该图的最小生成树是唯一的 ×

Ⅲ.用Prim算法从不同顶点开始构造的所有最小生成树一定相同 ×

Ⅳ.使用Prim和Kruskal算法得到的最小生成树总不相同 ×

A.仅Ⅰ　　B.仅Ⅱ　　C.仅Ⅰ、Ⅲ　　仅Ⅱ、Ⅳ

A

2 最短路径

❶ 单源最短路径—Dijkstra算法

源点*v*加入S，U=V-S
初始化：

从U中选择dist最小的顶点*u*

考察所有从*u*有出边的顶点*j*，调整：

若*v* →*i*有边：
　dist[*i*]=(*v*，*i*)权值
　path[*i*]=*v*
否则：
　dist[*i*]= ∞
　path[*i*]=-1

若dist[*u*]+(*u*，*j*)权值<dist[*j*]：
　dist[*j*]= dist[*u*]+(*u*，*j*)权值
　path[*j*]=*u*
否则：
　不变

直到S=V　时间复杂度：$O(n^2)$

Dijkstra算法是（ ）方法求出图中从源点到其余顶点最短路径的。

A.按长度递减的顺序求出图的某顶点到其余顶点的最短路径

B.按长度递增的顺序求出图的某顶点到其余顶点的最短路径 √

C.通过深度优先遍历求出图中某顶点到其余顶点的最短路径

D.通过广度优先遍历求出图中某顶点到其余顶点的最短路径

加入S集合的顶点：

- 最短路径不再改变
- 越后加入的顶点，dist越长

以下叙述正确的是（ ）。

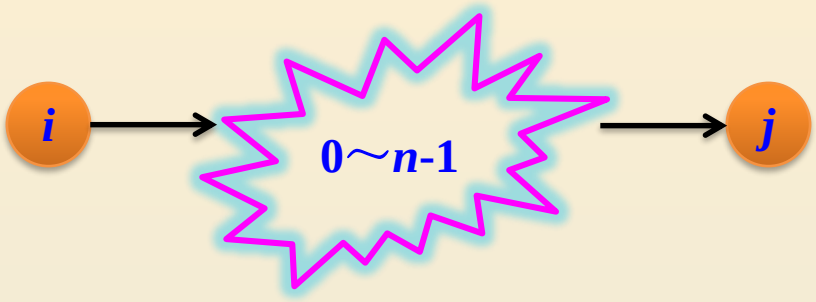
A. 最短路径一定是简单路径 √

B. Dijkstra算法不适合有回路的带权图求最短路径

C. Dijkstra算法不适合求任意两个顶点的最短路径

D. Floyd算法求两个顶点的最短路径时，$path_{k-1}$一定是$path_k$的子集

❷ 多源最短路径—Flody算法



迭代　时间复杂度：$O(n^3)$

Dijkstra算法用于求单源最短路径，为了求一个图中所有顶点对之间的最短路径，可以以每个顶点作为起点调用Dijkstra算法，Floyd算法和这种算法相比，有什么优势？

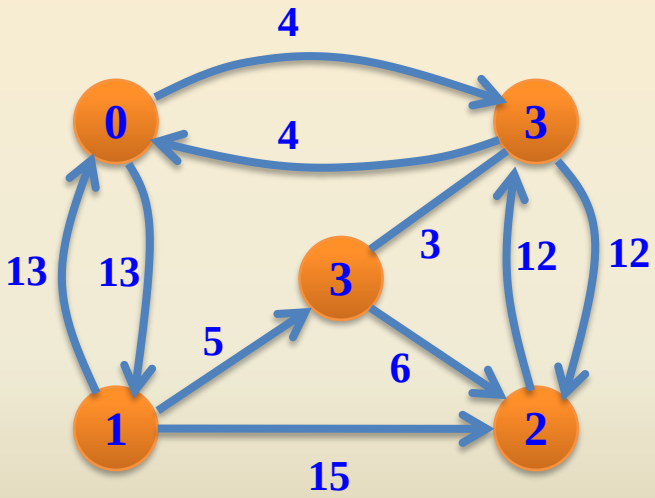
- 从每个顶点调用Dijkstra算法
- Floyd算法

} 时间复杂度：$O(n^3)$

- 从每个顶点调用Dijkstra算法：独立
- Floyd算法：A共享

⇒ Floyd算法性能更好

设下图中的顶点表示村庄，有向边代表交通路线，若要建立一家医院，试问建在哪一个村庄能使各村庄总体交通代价最小。



利用Floyd算法任意两个顶点之间的最短路径长度

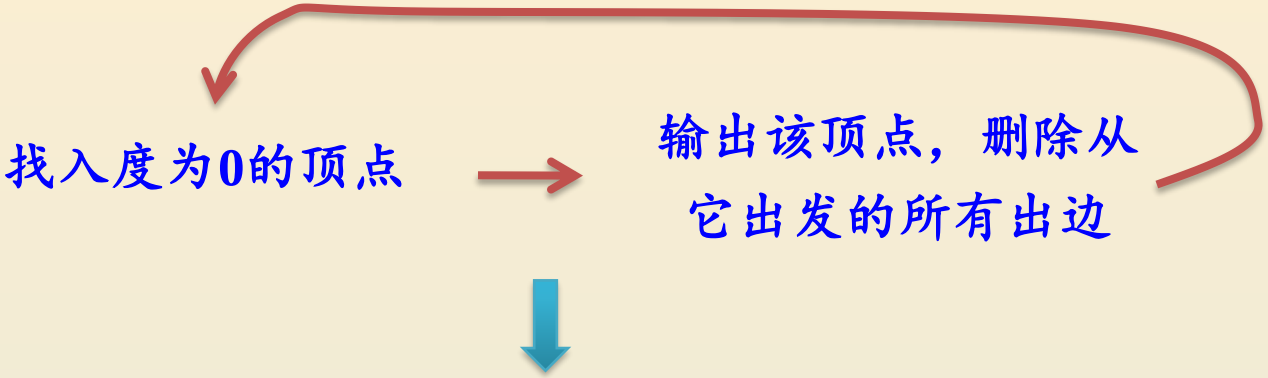
$$A_4 = \begin{bmatrix} 0 & 13 & 16 & 4 & 18 \\ 12 & 0 & 11 & 8 & 5 \\ 16 & 29 & 0 & 12 & 34 \\ 4 & 17 & 12 & 0 & 22 \\ 7 & 20 & 6 & 3 & 0 \end{bmatrix}$$

求得每对村庄之间的最少交通代价

医院建在的村庄	各村庄往返总的交通代价
0	12+16+4+7+13+16+4+18=90
1	13+29+17+20+12++8+5=115
2	16+11+12+6+16+29+12+34=136
3	4+8+12+3+4+17+12+22=**82**
4	18+5+34+22+7+20+6+3+0=115

把医院建在村庄3时总体交通代价最少。

3 拓扑排序



- **成功**：产生所有顶点的拓扑序列
- **不成功**：不能产生所有顶点的拓扑序列

若一个有向图中的顶点不能排成一个拓扑序列，则可断定该有向图（ ）

A.是个有根有向图

B.是个强连通图

C.含有多个入度为0的顶点

D.含有顶点数目大于1的强连通分量 √

不能排成一个拓扑序列 → 有回路 → 顶点数大于1的强连通分量

若用邻接矩阵存储有向图，矩阵中主对角线以下的元素均为零，则关于该图拓扑序列的结论是（ ）。

A.存在，且唯一　　B.存在、且不唯一

C.存在，可能不唯一 √　　D.无法确定是否存在

有向图：顶点$i \rightarrow j$（$i<j$）可能有边，而顶点$j \rightarrow i$一定没有边

↓

该有向图中一定没有回路

↓

可以产生拓扑序列，但拓扑序列不一定唯一

4 关键路径

- 对事件（顶点）进行拓扑排序
- 按拓扑序列求所有事件的最早开始时间
- 按拓扑逆序列求所有事件的最迟开始时间
- 求所有活动（边）的最早开始时间
- 求所有活动的最迟开始时间
- 关键活动：最早开始时间=最迟开始时间

以下对于AOE网的叙述中，错误的是（ ）。

A.在AOE网中可能存在多条关键路径 √

B.关键活动不按期完成就会影响整个工程的完成时间 √

C.任何一个关键活动提前完成，整个工程也将提前完成 ×

D.所有关键活动都提前完成，整个工程也将提前完成 √