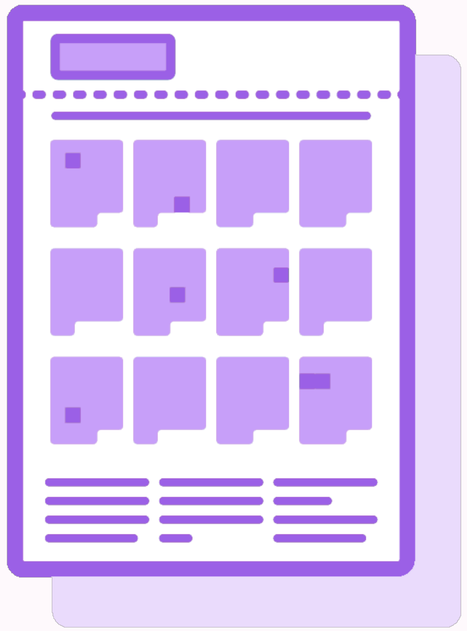
图

斐多课堂 Phaedo Classes 数据结构 不挂科 第六讲



3大模块

9道题目

数据结构—图

模块1／ 图的基本概念

模块2／ 图的遍历

模块3／图的基本应用

图的基本概念

小节1 / 定义

小节2 / 基本概念和术语

小节3 / 图结构的存储

图的基本概念

图的定义

图由结点的有穷组合V和边的集合E组，记为G=(V, E)。

图不可以是空图

图的基本概念

小节1 / 定义

小节2 / 基本概念和术语

小节3 / 图结构的存储

图的基本概念和术语

有向图：每条边都有方向

无向图：每条边都没有方向

有向图

无向图

图的基本概念和术语

有向完全图

在有向图中有n个顶点，将具有n(n-1)条边的有向图称为有向完全图。

图中任意两个顶点都有两条边相连。

无向完全图

在无向图中有n个顶点，将具有n(n-1)/2条边的无向图称为无向完全图。

图中任意两个顶点都有一条边相连。

图的基本概念和术语

连通

在无向图中，若从顶点v到顶点w有路径存在，则称v和w是连通的。

连通图

若图G中任意两个顶点都是连通的，则称图G为连通图，否则称为非连通图。

连通分量

无向图中的极大连通子图称为连通分量。

图的基本概念和术语

强连通

在有向图中，若从顶点v到顶点w和从顶点w到顶点v之间都有路径，则称这两个顶点是强连通的。

强连通图

若图中任何一对顶点都是强连通的，则称此图为强连通图。

强连通分量

有向图中的极大连通子图称为有向图的强连通分量。

图的基本概念和术语

弧

在有向图中，通常将边称为弧，含箭头的一端称为弧头，另一端称为弧尾

$<v_i, v_j>$表示从顶点v_i到顶点v_j有一条边

图的基本概念和术语

顶点的度

以该顶点为一个端点的边的数目。

顶点的入度

在有向图中，入度是以顶点v为终点的有向边的数目。

顶点的出度

在有向图中，出度是以顶点v为起点的有向边的数目。

有向图的全部顶点的入度之和与出度之和相等，并且等于边数。

图的基本概念和术语

路径

顶点v_p到顶点v_q之间的一条路径是指顶点序列v_p，v_{i1}，v_{i2}，…，v_{im}，v_q。

路径长度

路径上边的数目称为路径的长度。

回路

第一个顶点和最后一个顶点相同的路径称为回路或环。

如果一个图有n个顶点，并且有大于n-1条边，则此图一定有环。

图的基本概念和术语

简单路径

在路径序列中，顶点不重复出现的路径称为简单路径。

简单回路

除第一个顶点和最后一个顶点之外，其余顶点不重复出现的回路称为简单回路。

图的基本概念和术语

边的权

在一个图中，每条边都可以标上具有某种含义的数值，该数值称为该边的权值。

网

边上带有权值的图称为带权图，也称作网。

例题6-1

一个有n个顶点和n条边的无向图一定是（ ）。

A. 连通的　　B. 不连通的

C. 无环的　　D. 有环的

解析6-1

D

若一个无向图有n个顶点和n-1条边，可以使它连通但没有环（即生成树），但再加一条边，在不考虑重边的情形下，则必然会构成环。

例题6-2

若从无向图的任意顶点出发进行一次深度优先搜索即可访问所有顶点，则该图一定是（ ）。

A. 强连通路　　B. 连通路

C. 有回路　　D. 一棵树

解析6-2

B

A.强连通路为有向图，与题意矛盾。

B.对无向连通图做一次深度优先搜索，可以访问到该连通图的所有顶点。

C.有回路的无向图不一定是连通路，因为回路不一定包含图的所有结点。

D.连通图可能是树，也可能存在环。

图的基本概念

小节1 / 定义

小节2 / 基本概念和术语

小节3 / 图结构的存储

邻接矩阵法

用一个一维数组存储图中顶点的信息，用一个二维数组存储图中边的信息，存储顶点之间邻接关系的二维数组称为邻接矩阵。

设G=(V，E)是具有n个顶点的图，顶点序号依次为0，1，...，n-1，则G的邻接矩阵是具有如下定义的m阶方阵A：

A[i][j]=1 表示顶点i与顶点j邻接，即i与j之间存在边或者弧。

A[i][j]=0 表示顶点i与顶点j不邻接（0<=i, j<=n-1）。

邻接矩阵的结构体定义

```c
#define MAXVEX 100/* 最大顶点数，应由用户定义 */
#define INFINITY  65535 /* 表示权值的无穷*/

typedef int EdgeType;/* 边上的权值类型应由用户定义 */
typedef char VertexType;/* 顶点类型应由用户定义   */

typedef struct
{
    VertexType vexs[MAXVEX];/* 顶点表 */
    EdgeType arc[MAXVEX][MAXVEX];/* 邻接矩阵，可看作边表 */
    int numNodes, numEdges;/* 图中当前的顶点数和边数   */
} MGraph;
```

邻接表法

对图中的每个顶点i建立一个单链表，每个单链表的第一个结点存放有关顶点的信息，把这一结点看作链表的表头，其余结点存放有关边的信息。

邻接表由单链表的表头形成的顶点表和单链表其余结点形成的边表两部分组成。

一般顶点表存放顶点信息和指向第一个边结点指针，边表存放与当前顶点相邻接顶点的序号和指向下一个边结点的指针。

图的遍历

小节1 / 深度优先搜索遍历

小节2 / 广度优先搜索遍历

图的遍历

小节1 / 深度优先搜索遍历

小节2 / 广度优先搜索遍历

深度优先搜索遍历DFS的基本思想

深度优先搜索遍历（DFS）类似于二叉树的先序遍历。

1. 访问出发点v，并将其标记为已访问过
2. 选取与v邻接的未被访问的任意一个顶点w，并访问它
3. 选取与w邻接的未被访问的任一顶点并访问，以此重复进行

当一个顶点所有的邻接顶点都被访问过时，则依次退回到最近被访问过的顶点

若该顶点还有其他邻接顶点未被访问，则从这些未被访问的顶点中取一个并重复上述访问过程，直至图中所有顶点都被访问过为止

深度优先搜索遍历

深度优先搜索遍历

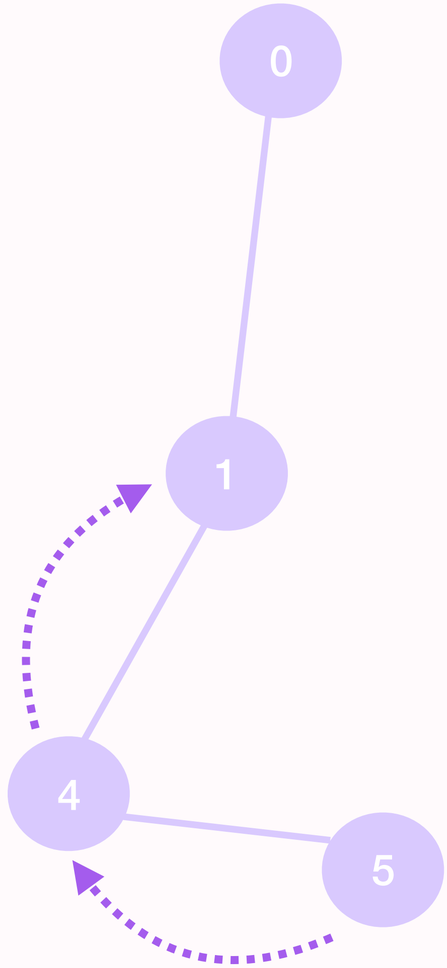
访问指定的起始顶点；若当前访问的顶点的邻接顶点有未被访问的，则任选一个访问之；

深度优先搜索遍历

访问指定的起始顶点；若当前访问的顶点的邻接顶点有未被访问的，则任选一个访问之；

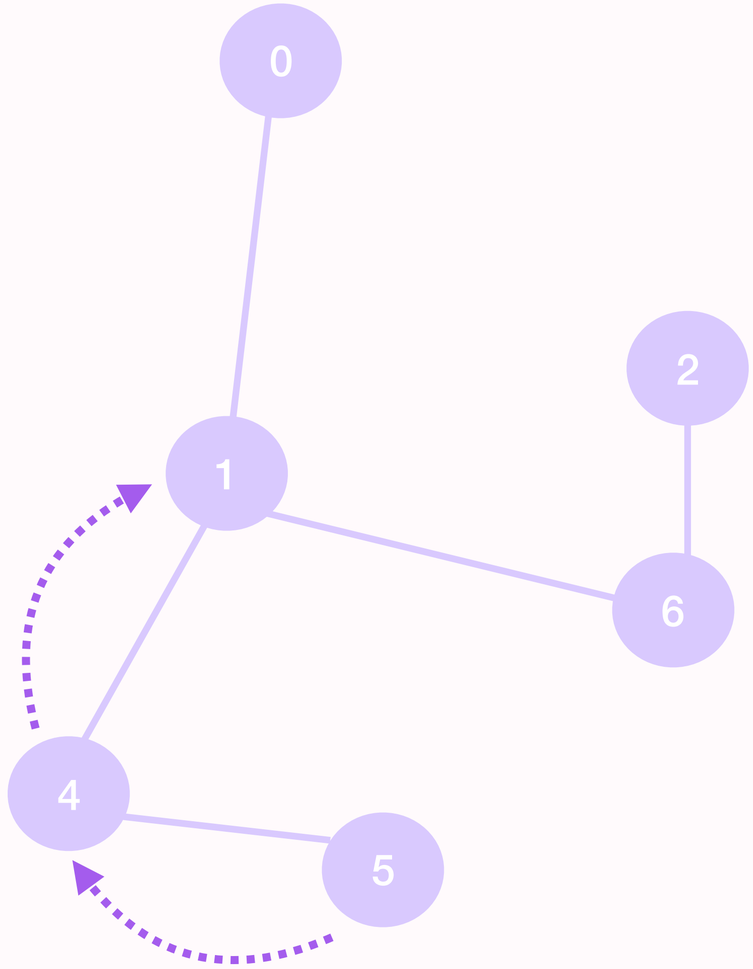
深度优先搜索遍历

反之，退回到最近访问过的顶点；直到与起始顶点相通的全部顶点都访问完毕；



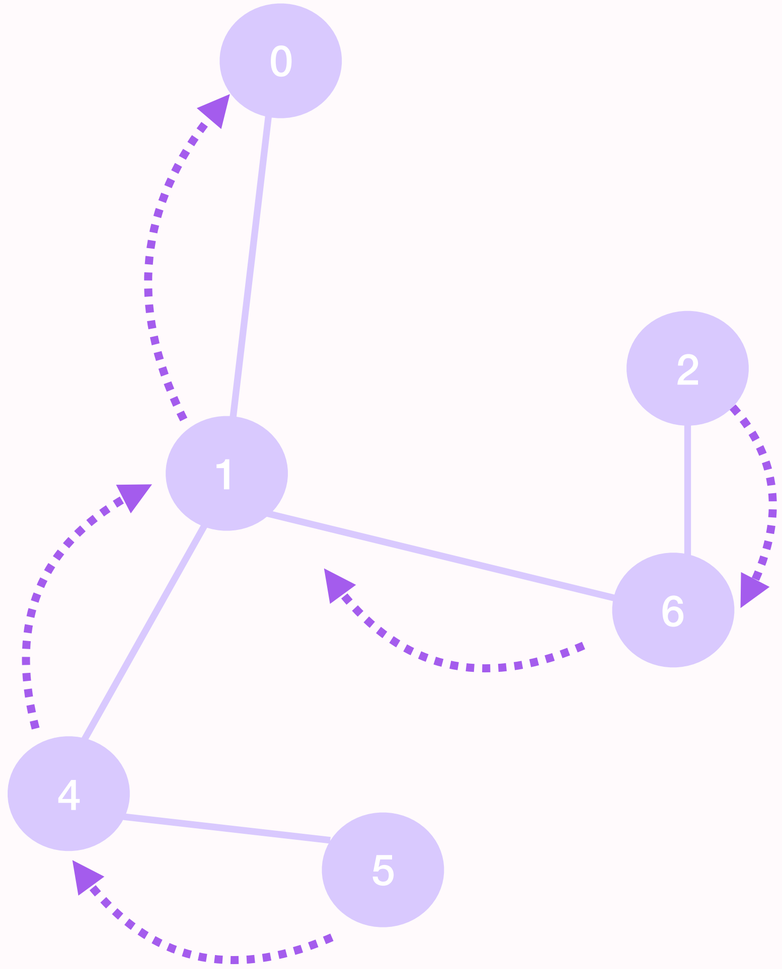
深度优先搜索遍历

回退到1，发现了新的没有被访问的结点



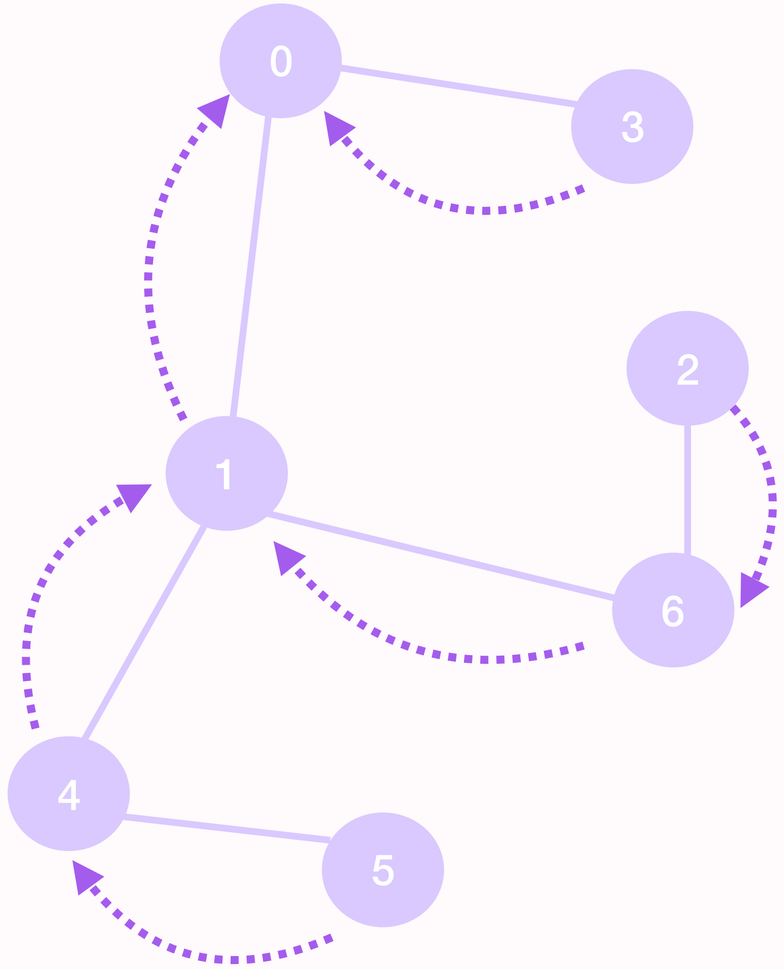
深度优先搜索遍历

继续回退，回退到0



深度优先搜索遍历

再也找不到新的结点了，那么回退，回退到起始顶点，结束搜索



深度优先搜索遍历DFS的性能分析

是一个递归算法，需要借助一个递归工作栈，时间复杂度为O(|V|)。

以邻接矩阵表示时，查找每个顶点的邻接点所需时间为O(|V|)，总的时间复杂度为$O(|V|^2)$。

以邻接表表示时，查找所有顶点的邻接点所需时间为O(|E|)，访问顶点所需时间为O(|V|)，
总的时间复杂度为O(|V|+|E|)。

深度优先搜索树

把图的深度优先搜索遍历过程中所经历的边保留，其余的边删掉，形成的树称为深度优先搜索生成树。

图的遍历

小节1 / 深度优先搜索遍历

小节2 / 广度优先搜索遍历

广度优先搜索遍历BFS的基本思想

图的广度优先搜索遍历（BFS）类似于树的层次遍历。

1. 访问起始点v
2. 选取与v邻接的全部顶点w_1，w_2，...，w_n进行访问
3. 再依次访问与w_1，w_2，...，w_n邻接的全部顶点（已经访问过的除外），依此类推，直到所有顶点都被访问过

广度优先搜索遍历BFS的算法过程

方法：从图的某一结点出发，首先依次访问该结点的所有邻接顶点 Vi1, Vi2, …, Vin 再按这些顶点被访问的先后次序依次访问与它们相邻接的所有未被访问的顶点，重复此过程，直至所有顶点均被访问为止。

0 1 2 3 4 5 6

广度优先搜索遍历BFS的性能分析

BFS邻接表和邻接矩阵的存储方式都需要借助一个辅助队列Q，n个顶点均需入队一次，最坏的情况下空间复杂度为O(|V|)。

采用邻接表存储方式时，每个顶点均需搜索一次，时间复杂度为O(|V|)。

在搜索任一顶点时，每条边至少访问一次，时间复杂度为O(|E|)。

总时间复杂度为O(|V|)+|E|)。

采用邻接矩阵存储方式时，查找每个顶点所需时间为O(|V|)，总时间复杂度为$O(|V|^2)$。

广度优先搜索树

一给定的邻接矩阵存储表示是唯一的，故其广度优先生成树也是唯一的。

但邻接表存储表示不唯一，故其广度优先生成树不唯一。

例题6-3

下列关于广度优先算法点说法中，正确的是（ ）。（多选）

A. 当各边的权值相等时，广度优先算法可以解决单源最短路径问题
B. 当各边的权值不等时，广度优先算法可用来解决单源最短路径问题
C. 广度优先遍历算法类似于树中的后序遍历算法
D. 实现图的广度优先算法时，使用的数据结构是队列

解析6-3

AD

A、B.广度优先搜索以起始结点为中心，一层一层地向外层扩展遍历图的顶点，因此无法考虑到边的权值，只适合求边权值相等的图的单源最短路径。
C.广度优先搜索相当于树的层序遍历。
D.广度优先搜索需要用到队列，深度优先搜索需要用到栈。

例题6-4

对一个有n个顶点、e条边的图采用邻接表表示时，进行DFS遍历的时间复杂度为（ ），空间复杂度为（ ）；进行BFS遍历的时间复杂度为（ ），空间复杂度（ ）。

A. O（n）
B. O（e）
C. O（n+e）
D. O（1）

解析6-4

C、A、C、A

深度优先遍历时，每个顶点表结点和每个边表结点均查询一次，每个顶点递归调用一次，需要借助一个递归工作栈；而广度优先遍历时，也是每个顶点表结点和每个边表结点均查找一次，每个顶点进入队列一次。

图的基本应用

小节1 / 最小生成树

小节2 / 最短路径

小节3 / 拓扑排序与关键路径

图的基本应用

最小生成树的定义

设R为G的所有生成树的集合，若T为R中边的权值之和最小的那棵生成树，则称T为G的最小生成树。

最小生成树的性质

最小生成树不是唯一的

最小生成树的边的权值之和总是唯一的

最小生成树的边数为顶点数减1

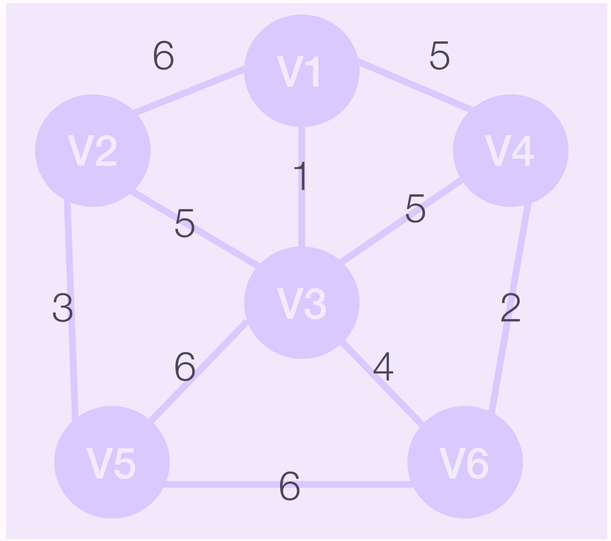
普里姆Prim算法的思想

1. 从图中任意取出一个顶点，把它当成一棵树

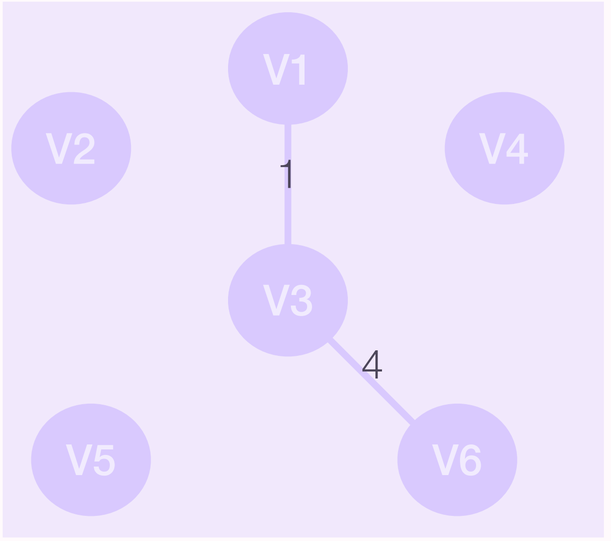
2. 从与这棵树相接的边中选取一条最短的边，并将这条边及其所连接的顶点也并入这棵树中，得到一棵两个顶点的树

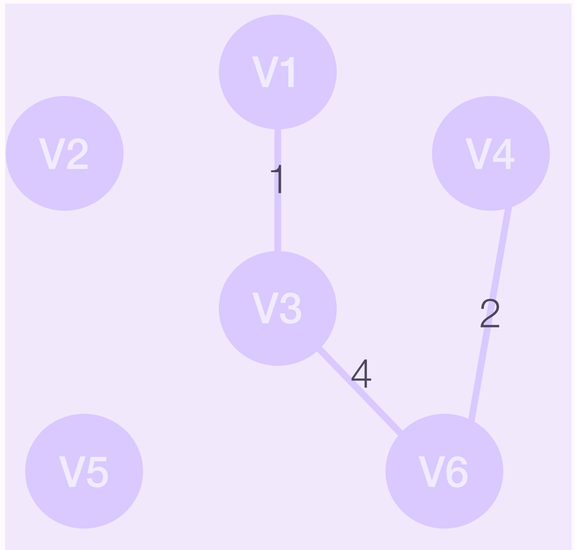
3. 从与这棵树相接的边中选取一条最短的边，并将这条边及其所连顶点并入当前树中，依次类推，直到图中所有顶点都被并入树中为止

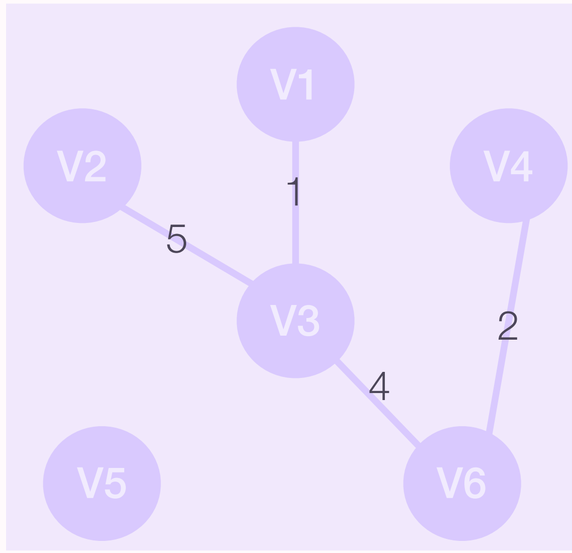
普里姆Prim算法

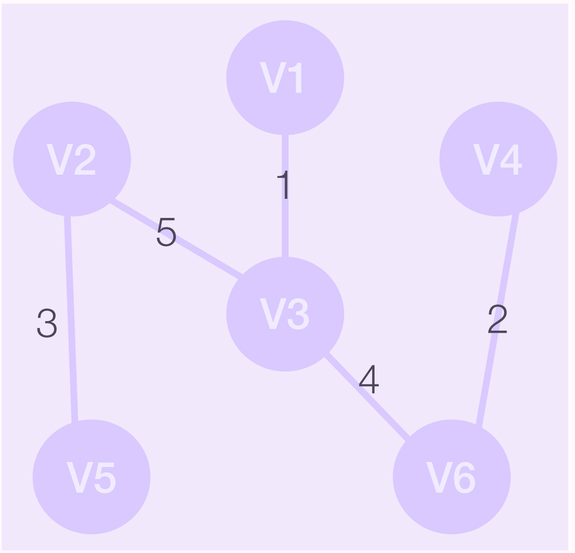

V1
V2
V3
V4
V5
V6
6
5
1
5
5
3
2
6
4
6

V1
V2
V3
V4
V5
V6
1


V1
V2
V3
V4
V5
V6
1
4


V1
V2
V3
V4
V5
V6
1
2
4


V1
V2
V3
V4
V5
V6
1
5
2
4


V1
V2
V3
V4
V5
V6
1
5
3
2
4

普里姆Prim算法的性能分析

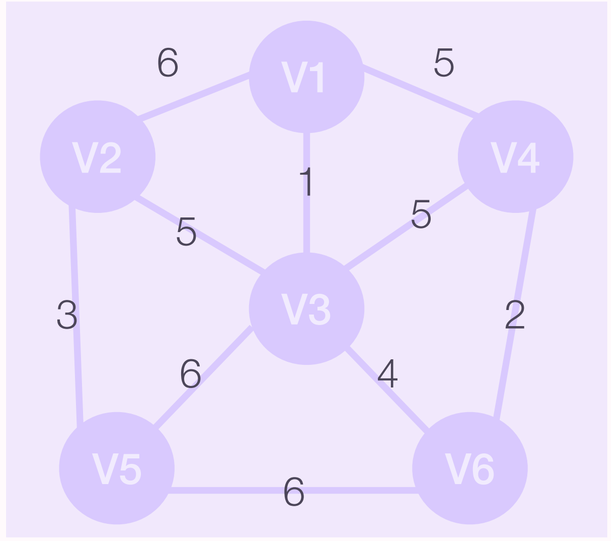
时间复杂度为$O(|V|^2)$

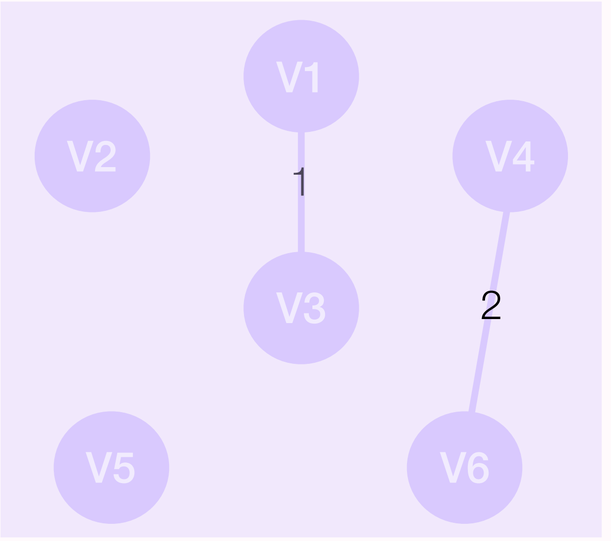
适用于稠密图

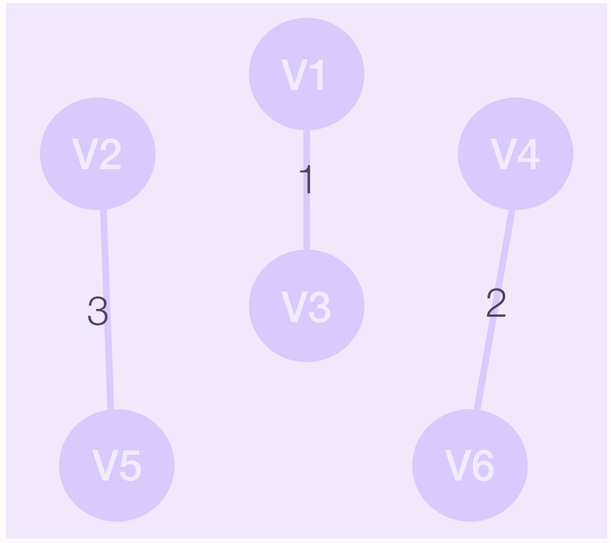
克鲁卡斯尔Kruskal算法的思想

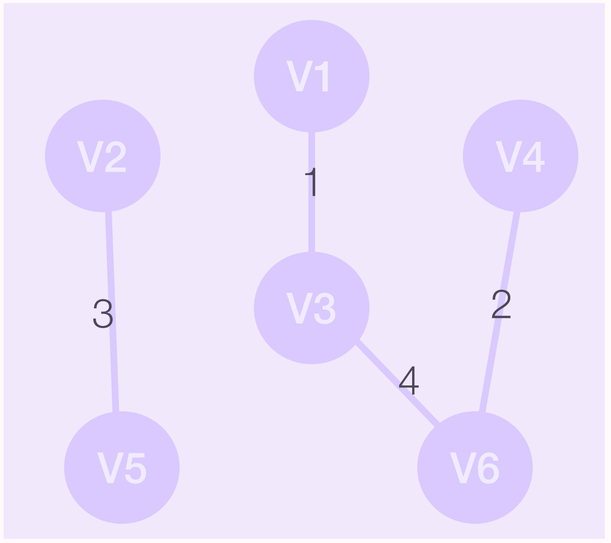
1. 每次找出候选边中权值最小的边，将该边并入生成树中
2. 重复上述过程直到所有边都被检测完为止

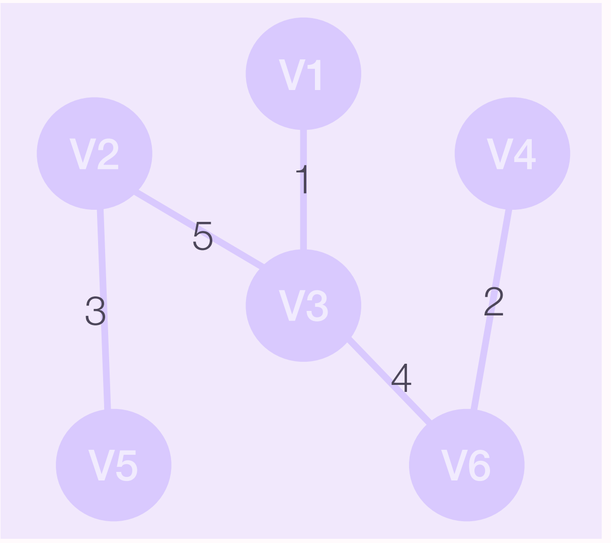
克鲁卡斯尔Kruskal算法











克鲁卡斯尔Kruskal算法的性能分析

时间复杂度由选取的排序算法决定

适用于稀疏图

例题6-5 用Pirm算法和Kruskal算法构造图的最小生成树，所得到的最小生成树（ ）。

A. 相同
B. 不相同
C. 可能相同，可能不同
D. 无法比较

解析6-5 **C**

由于无向连通图的最小生成树不一定唯一，所以用不同算法生成的最小生成树可能不同，但当无向连通图的最小生成树唯一时，不同算法生成的最小生成树必定是相同的。

例题6-6 任何一个无向连通图的最小生成树（ ）。

A. 有一棵或多棵
B. 只有一棵
C. 一定有多棵
D. 可能不存在

解析6-6 **A**

当无向连通图的存在权值相同的多条边时，最小生成树可能是不唯一的；另外，由于这是一个无向连通图，故最小生成树必定存在。

图的基本应用

小节1 / 最小生成树

小节2 / 最短路径

小节3 / 拓扑排序与关键路径

最短路径的定义

把带权路径长度最短的那条路径称作最短路径。

迪杰斯特拉Dijkstra算法的思想

设有两个顶点集合S和T，集合S中存放图中已找到最短路径的顶点，集合T中存放图中剩余顶点

初始状态时，集合S中只包含源点v_0

从集合T中选取到顶点v_0路径长度最短的顶点v_u并入到集合S中

集合S每并入一个新的顶点v_u，都要修改顶点v_0到集合T中顶点的最短路径长度值

不断重复此过程，直到集合T的顶点全部并入到S中为止

迪杰斯特拉Dijkstra算法的执行过程

引入三个辅助数组dist[ ], path[ ], set[ ]

dist[v_1]表示当前已找到的从v_0到每个终点v_1的最短路径的长度。初态为：若从v_0到v_1有边，则dist[v_1]为边上的权值；否则为∞

path[v_i]中保存从v_0到v_i最短路径上v_i的前一个顶点。初态为：若从v_0到v_i有边，则path[v_i]=v_0；否则path[v_i]=-1

set[ ]为标记数组，set[v_i]=0表示v_i在T中，没有被并入最短路径；set[v_i]=1表示v_i在S中，即已经被并入最短路径。初态为：set[v_0]=1，其余元素全为0

迪杰斯特拉Dijkstra算法的性能分析

基于贪心策略，主要部分为一个双重循环，外层循环内有两个并列的单层循环

时间复杂度为$O(n^2)$

利用Dijkstra算法不一定能得到正确的结果

弗洛伊德Floyd算法的基本思想

弗洛伊德算法定义了两个二维矩阵：
矩阵D记录顶点间的最小路径

例如D[0][3]= 10，说明顶点0 到 3 的最短路径为10；
矩阵P记录顶点间最小路径中的中转点

例如P[0][3]= 1 说明，0 到 3的最短路径轨迹为：0 -> 1 -> 3。

它通过3重循环，k为中转点，v为起点，w为终点，循环比较D[v][w] 和 D[v][k] + D[k][w] 最小值，
如果D[v][k] + D[k][w] 为更小值，则把D[v][k] + D[k][w] 覆盖保存在D[v][w]中。

弗洛伊德Floyd算法的性能分析

计算图中任意一对顶点间的最短路径

主要部分是一个三层循环，取内层循环的操作为基本操作，基本操作执行次数为n^3，
时间复杂度为$O(n^3)$

图的基本应用

拓扑排序中的基本概念

有向无环图

一个有向图中不存在环，则称为有向无环图，简称DAG图

AOV网

活动在顶点上的网（AOV）是一种可以形象地反映出整个工程中各个活动之间的先后关系的有向图

拓扑排序

在图论中，由一个有向无环图的顶点组成的序列满足下列条件：

1. 每个顶点出现且只出现一次

2. 若顶点A在序列中排在顶点B的前面，则在图中不存在从顶点B到顶点A的路径

拓扑排序中的常用步骤

1. 从DAG图中选择有一个没有前驱的顶点并输出
2. 从图中删除该顶点和所有以它为起点的有向边
3. 重复1和2直到当前的DAG图为空，或当前图中不存在无前驱的顶点为止

关键路径中的基本概念

AOE网	在带权有向图中，以顶点表示事件，有向边表示活动，边上的权值表示完成该活动的开销，则称这种有向图为活动在边上的网（AOE）
关键路径	从源点到汇点的所有路径中，具有最大路径长度的路径称为关键路径
关键活动	把关键路径上的活动称为关键活动

关键路径中的参量

事件v_k的最早发生时间ve(k)

从开始顶点V到Vk的最长路径长度

ve(源点) = 0

$ve(k) = Max\{ve(j) + Weight(v_j, v_k)\}$

关键路径中的参量

事件v_k的最迟发生时间vl(k)

保证它所指向的事件v_i在ve(i)时刻能够发生时，该事件最迟必须发生的时间

vl(汇点) = ve(汇点)

$vl(j) = Min\{vl(k) - Weight(v_j, v_k)\}$

关键路径中的参量

活动a_i的最早开始时间e(i)

该活动的起点所表示的事件最早发生时间

e(i) = ve(k)

关键路径中的参量

活动a_i的最迟开始时间l(i)

该活动的终点所表示的事件最迟发生时间与该活动所需时间之差

$$l(i) = vl(j) - Weight(v_k, v_j)$$

关键路径中的参量

活动a_i的最迟开始时间l(i)和其最早开始时间e(i)的差额d(i)

该活动完成的时间余量，即在不增加完成整个工程所需的总时间的情况下，活动a_i可以拖延的时间

$$d(i) = l(i) - e(i)$$

例题6-7 若一个有向图的顶点不能排在一个拓扑序列中，则可判定该有向图（ ）。

A. 是一个有根的有向图
B. 是一个强连通图
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

解析6-7 **D**

若不存在拓扑排序，则表示图中必定存在回路，该回路构成一个强连通分量（也可理解为顶点数目大于1的强连通分量中必然存在回路）。

例题6-8

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

解析6-8

C

对角线以下的元素均为零，表明只有从顶点i到顶点j（i<j）可能有边，而从顶点j到顶点i一定无边，即有向图是一个无环图，因此一定存在拓扑序列。对于拓扑序列是否唯一，试举一例：设有向图的邻接矩阵为：

0	1	1
0	0	0
0	0	0

则存在两个拓扑序列，因此该图存在可能不唯一的拓扑序列。若题目中说对角线以上的元素均为1，以下的元素均为0，则拓扑序列唯一。

例题6-9 若一个有向图具有有序的拓扑排序序列，则它的邻接矩阵必定为（ ）。

A. 对称
B. 稀疏
C. 三角
D. 一般

解析6-9 **C**

这道题争议较大，因为在书中漏掉了“有序”二字。可以证明，对有向图中的顶点适当地编号，使其邻接矩阵为三角矩阵且主对角元全为零的充分必要条件是，该有向图可以进行拓扑排序。若这道题将“有序”二字去掉，则应该选D。
需要注意的是，若一个有向图的邻接矩阵为三角矩阵（对角线上元素为0），则图中必不存在环，因此其拓扑序列必然存在。

图

斐多课堂 Phaedo Classes

数据结构 不挂科 第六讲