

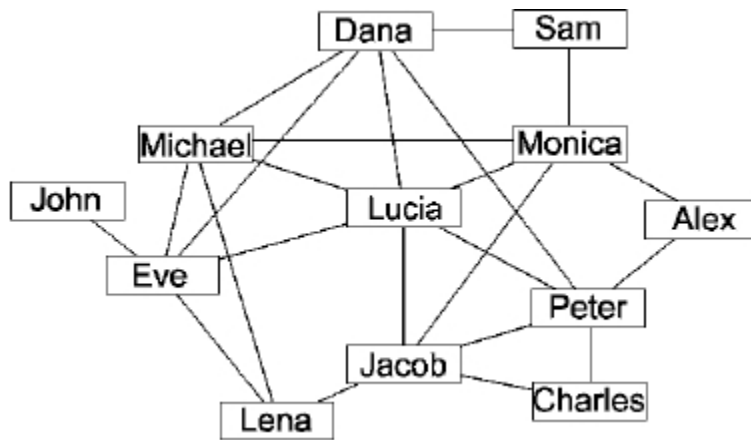


# 【课堂练习】



## 课堂练习

1. 【NOIP2016】Lucia 和她的朋友以及朋友的朋友都在某社交网站上注册了账号。下图是他们之间的关系图，两个人之间有边相连代表这两个人是朋友，没有边相连代表不是朋友。这个社交网站的规则是：如果某人 A 向他（她）的朋友 B 分享了某张照片，那么 B 就可以对该照片进行评论；如果 B 评论了该照片，那么他（她）的所有朋友都可以看见这个评论以及被评论的照片，但是不能对该照片进行评论（除非 A 也向他（她）分享了该照片）。现在 Lucia 已经上传了一张照片，但是她不想让 Jacob 看见这张照片，那么她可以向以下朋友( )分享该照片。



A. Dana, Michael, Eve

B. Dana, Eve, Monica

C. Michael, Eve, Jacob

D. Micheal, Peter, Monica

## 课堂练习

2. 【NOIP2014】有向图中每个顶点的度等于该顶点的( )。
- A.入度                      B.出度                      C.入度与出度之和                      D.入度与出度之差
3. 【NOIP2014】在无向图中，所有顶点的度数之和是边数的( )倍。
- A.0.5                      B.1                      C.2                      D.4
4. 【NOIP2016】设简单无向图  $G$  有 16 条边且每个顶点的度数都是 2，则图  $G$  有( )个顶点。
- A. 10                      B. 12                      C. 8                      D. 16
5. 【NOIP2010】关于拓扑排序，下面说法正确的是( )。
- A.所有连通的有向图都可以实现拓扑排序
- B.对一个图而言，拓扑排序的结果是唯一的
- C.拓扑排序中入度为 0 的结点总会排在入度大于 0 的结点的前面
- D.拓扑排序结果序列中的第一个结点一定是入度为 0 的点

6. 【NOIP2008】设  $T$  是一棵有  $n$  个顶点的树，下列说法不正确的是 ( )。
- A.  $T$  有  $n$  条边  
B.  $T$  是连通的  
C.  $T$  是无环的  
D.  $T$  有  $n-1$  条边
7. 【NOIP2017】设  $G$  是有  $n$  个结点、 $m$  条边 ( $n \leq m$ ) 的连通图，必须删去  $G$  的 ( ) 条边，才能使得  $G$  变成一棵树。
- A.  $m - n + 1$       B.  $m - n$       C.  $m + n + 1$       D.  $n - m + 1$
8. 【NOIP2015】6 个顶点的连通图的最小生成树，其边数为( )。
- A.6      B.5      C.7      D.4
9. 【NOIP2014】设  $G$  是有 6 个结点的完全图，要得到一棵生成树，需要从  $G$  中删去( ) 条边。
- A.6      B.9      C.10      D.15

10. 【NOIP2009】已知  $n$  个顶点的有向图，若该图是强连通的（从所有顶点都存在路径到达其他顶点），则该图中最少有多少条有向边？

A.  $n$ B.  $n+1$ C.  $n-1$ D.  $n*(n-1)$ 

11. 【NOIP2011】无向完全图是图中每对顶点之间都恰有一条边的简单图。已知无向完全图  $G$  有 7 个顶点，则它共有( )条边。

A. 7

B. 21

C. 42

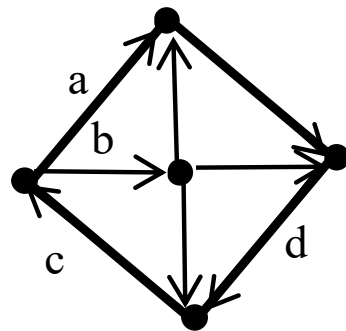
D. 49

12. 【NOIP2011】对一个有向图而言，如果每个节点都存在到达其他任何节点的路径，那么就称它是强连通的。例如，右图就是一个强连通图。事实上，在删掉边( )后，它依然是强连通的。

A. a

B. b

C. c



13. 【NOIP2013】在一个无向图中，如果任意两点之间都存在路径相连，则称其为连通图。

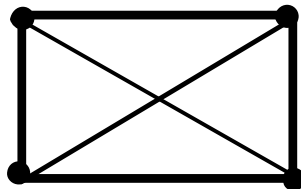
下图是一个有 4 个顶点、6 条边的连通图。若要使它不再是连通图，至少要删去其中的( ) 条边。

A.1

B.2

C.3

D.4



14. 【NOIP2013】在一个无向图中，如果任意两点之间都存在路径相连，则称其为连通图。

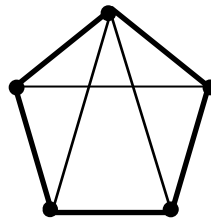
右图是一个有 5 个顶点、8 条边的连通图。若要使它不再是连通图，至少要删去其中的( ) 条边。

A.2

B.3

C.4

D.5



15. 【NOIP2013】二分图是指能将顶点划分成两个部分，每一部分内的顶点间没有边相连的简单无向图。那么，12 个顶点的二分图至多有( )条边。

A.18

B.24

C.36

D.66

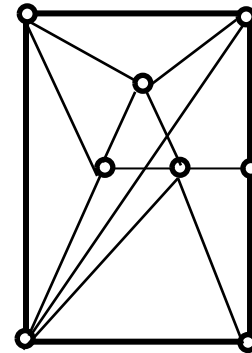
16. 【NOIP2015】对图  $G$  中各个结点分别指定一种颜色，使相邻结点颜色不同，则称为图  $G$  的一个正常着色。正常着色图  $G$  所必需的最少颜色数，称为  $G$  的色数。那么下图的色数是( )。

A.3

B.4

C.5

D.6



17. 【NOIP2016】 $G$  是一个非连通简单无向图，共有 28 条边，则该图至少有 ( ) 个顶点。

A. 10

B. 9

C. 8

D. 7

18. 【NOIP2017】由四个不同的点构成的简单无向连通图的个数是 ( )。

A. 32

B. 35

C. 38

D. 41

19. 【NOIP2018】由四个没有区别的点构成的简单无向连通图的个数是 ( )。

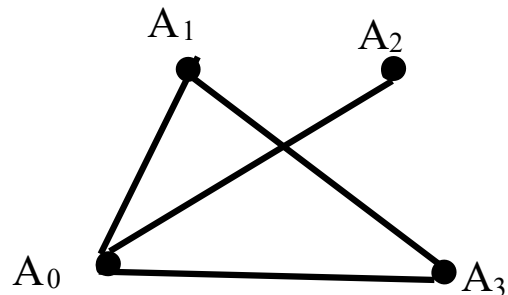
A.6

B.7

C.8

D.9

20. 【NOIP2013】以  $A_0$  作为起点，对下面的无向图进行深度优先遍历时，遍历顺序不可能的是( )。



A.  $A_0, A_1, A_2, A_3$    B.  $A_0, A_1, A_3, A_2$    C.  $A_0, A_2, A_1, A_3$    D.  $A_0, A_3, A_1, A_2$

21. 【NOIP2015】具有  $n$  个顶点， $e$  条边的图采用邻接表存储结构，进行深度优先遍历和广度优先遍历运算的时间复杂度均为( )。

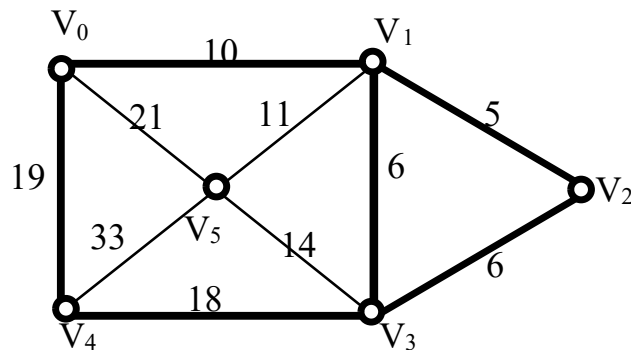
A.  $O(n^2)$                       B.  $O(e^2)$                       C.  $O(ne)$                       D.  $O(n + e)$

22. 【NOIP2013】对一个  $n$  个顶点、 $m$  条边的带权有向简单图用 Dijkstra 算法计算单源最短路时，如果不使用堆或其它优先队列进行优化，则其时间复杂度为( )。

A.  $O(mn + n^3)$                       B.  $O(n^2)$   
C.  $O((m + n) \log n)$                       D.  $O((m + n^2) \log n)$

23. 【NOIP2009】右图给出了一个加权无向图，从顶点  $V_0$  开始用 **prim** 算法求最小生成树。则依次加入最小生成树的顶点集合的顶点序列为：

- A.  $V_0, V_1, V_2, V_3, V_5, V_4$
- B.  $V_0, V_1, V_5, V_4, V_3, V_3$
- C.  $V_1, V_2, V_3, V_0, V_5, V_4$
- D.  $V_1, V_2, V_3, V_0, V_4, V_5$





# 【不定项选择题】





## 课堂练习

4. 【NOIP2012】已知带权有向图  $G$  上的所有权值均为正整数，记顶点  $u$  到顶点  $v$  的最短路径的权值为  $d(u, v)$ 。若  $v_1, v_2, v_3, v_4, v_5$  是图  $G$  上的顶点，且它们之间两两都存在路径可达，则以下说法正确的有( )。

A.  $v_1$  到  $v_2$  的最短路径可能包含一个环

B.  $d(v_1, v_2) = d(v_2, v_1)$

C.  $d(v_1, v_3) \leq d(v_1, v_2) + d(v_2, v_3)$

D. 如果  $v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4 \rightarrow v_5$  是  $v_1$  到  $v_5$  的一条最短路径，那么  $v_2 \rightarrow v_3 \rightarrow v_4$  是  $v_2$  到  $v_4$  的一条最短路径

5. 【NOIP2015】以下图中一定可以进行黑白染色的有( )。(黑白染色：为各个结点分别指定黑白两种颜色之一，使相邻结点颜色不同。)

A. 二分图

B. 完全图

C. 树

D. 连通图

6. 【NOIP2018】下列关于最短路算法的说法正确的有( )。

A. 当图中不存在负权回路但是存在负权边时，Dijkstra 算法不一定能求出源点到所有点的最短路。

B. 当图中不存在负权边时，调用多次 Dijkstra 算法能求出每对顶点间最短路径。

C. 图中存在负权回路时，调用一次 Dijkstra 算法也一定能求出源点到所有点的最短路。

D. 当图中不存在负权边时，调用一次 Dijkstra 算法不能用于每对顶点间最短路径计算

7. 【NOIP2011】对右图使用 Dijkstra 算法计算 S 点到其余各点的最短路径长度时，到 B 点的距离  $d[B]$  初始时赋为 8，在算法的执行过程中还会出现 的值有( )。

A. 3

B. 7

C. 6

D. 5

