# 习题4 图结构

4-1 设某个非连通无向图有25条边，问该图至少有( C )个顶点。

(A) 7

(B) 8

(C) 9

(D) 10

4-2 设某无向图中有n个顶点e条边，则建立该图邻接表的时间复杂度为( A )。

(A) O(n+e)

(B) O(n2)

(C) O(ne)

(D) O(n3)

4-3 带权有向图G用邻接矩阵R存储，则顶点i的入度等于R中( D  )。

(A) 第i行非∞(或非0)的元素之和

(B) 第i列非∞(或非0)的元素之和

(C) 第i行非∞(或非0)的元素个数

(D) 第i列非∞(或非0)的元素个数

4-4下面关于无向图的存储结构叙述中，正确的是( B )。

(A) 用邻接表存储图，占用的存储空间大小与图中边数有关，与顶点数无关

(B) 用邻接表存储图，占用的存储空间大小与图中边数和顶点数都有关

(C) 用邻接矩阵存储图，占用的存储空间大小与图中边数和顶点数都有关

(D) 用邻接矩阵存储图，占用的存储空间大小与图中边数有关，与顶点数无关

4-5 设图G=(V, E)，V={ a, b, c, d, e }，E={<a, b>, <a, c>, <b, d>, <c, e>, <d, c>, <e, d>}。

1. 是否存在从c到b的路径?

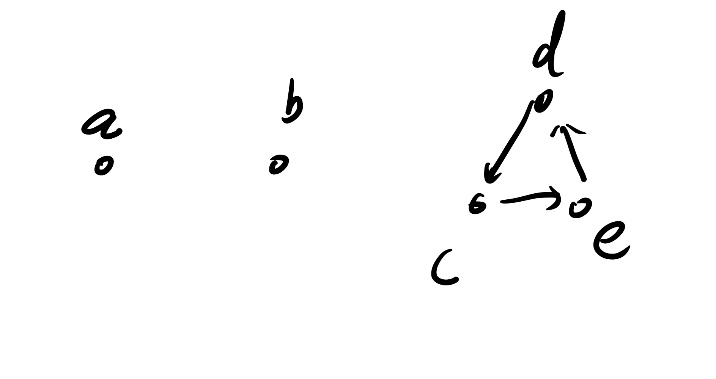
不存在

1. 计算ID(d)、OD(d)、TD(d)；

ID(d)=2

OD(d)=1

TD(d)=3

(3)画出各个强连通分量。

4-6 设计算法，由依次输入的顶点数目、弧的数目、各个顶点元素信息和各条弧信息建立有向图的邻接表。

void CreateGraph(graph\* &G) {

G = new graph;

cout << "输入顶点数和边数" << endl;

cin >> G->vexnum >> G->arcnum;

cout << "图表类型" << endl;

cin >> G->graphtype;

cout << "依次输入各顶点及其出边" << endl;

for (int i = 1; i <= G->vexnum; i++) {

cin >> G->vexlist[i].vex;

G->vexlist[i].firstarc = nullptr;

}

for (int i = 1; i <= G->arcnum; i++) {

int j;

int k;

cin >> j >> k;

arcnode\* temp = new arcnode;

temp->adjvex = k;

temp->next = G->vexlist[j].firstarc;

G->vexlist[j].firstarc = temp;

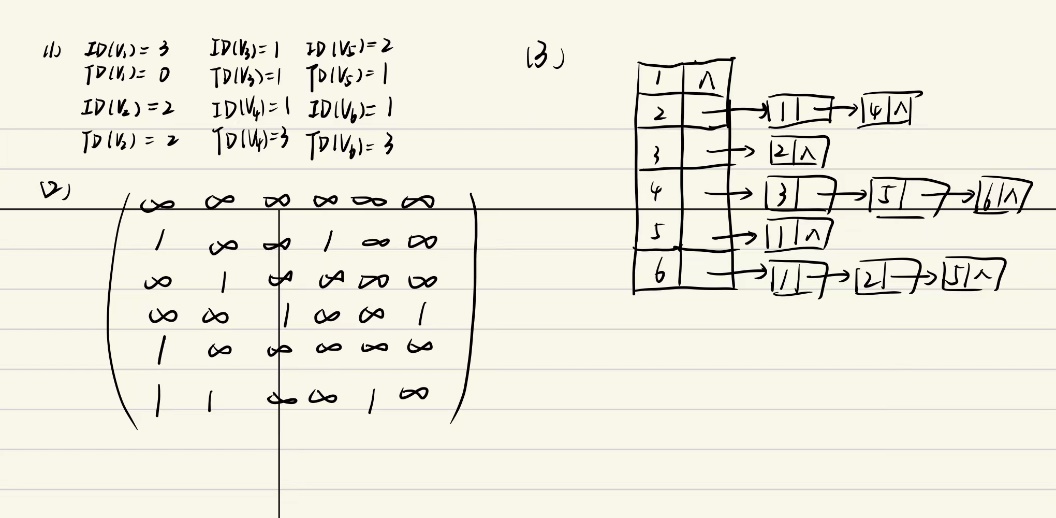
}//头插法插入各出边

return;

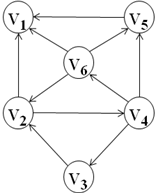
}

4-7 请给出有向图的

(1) 每个顶点的入度和出度；

(2) 邻接矩阵；

(3) 邻接表。



4-8 设无向图G=(V，E)，V={a，b，c，d，e，f}，E={(a，b)，(a，e)，(a，c)，(b，e)，(c，f)，(f，d)，(e，d)}。从顶点a出发对图G进行深度优先搜索遍历，得到的顶点序列是( D )。

(A) a b e c d f

(B) a c f e b d

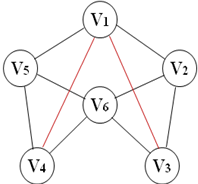
(C) a e b c f d

(D) a e d f c b

4-9 假设v1为出发点，优先考虑编号的顶点。试给出无向图的

(1)深度优先遍历的顶点序列和边序列；

(2)广度优先遍历的顶点序列和边序列。



1. 顶点序列:{1,2,3,6,4,5}

边序列:{(1,2)，(2,3)，(3,6)，(6,4)，(4,5)}

1. 顶点序列：{1，2，5，3，6，4}

边序列:{(1,2),(1,5),(2,3),(2,6),(5,4)}

4-10 概念解释：最小生成树。

一个有 n 个结点的连通图的最小生成树是原图的极小连通子图，且包含原图中的所有 n 个结点，并且有保持图连通的最少的边。可以由克鲁斯卡尔或普里姆算法得出

4-11 设无向图G=(V, E)，V={a, b, c, d, e}，E={<a, b>, <a, c>, <a, d>, <b, c>, <c, e>, <d, e>}，G1=(V, E1)。如果G1是G的生成树，则错误的是( D )。

(A) E1={<a, b>，<a, c>，<a, d>，<c, e>}

(B) E1={<a, b>，<a, c>，<c, e>，<d, e>}

(C) E1={<a, c>，<b, c>，<c, e>，<d, e>}

(D) E1={<a, d>，<b, c>，<c, d>，<d, e>}

4-12 判断一个有向图是否存在回路，除了可以利用深度优先遍历算法外，还可以利用( C )。

(A) 广度优先遍历算法

(B) 求最短路径的方法

(C) 拓扑排序方法

(D) 求关键路径的方法

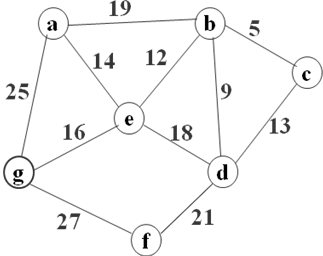
4-13 设带权无向图G =(V, E)含有n个顶点m条边。试描述构造图G的最小生成树的克鲁斯卡尔(Kruskal)算法。

1.构造一个只有 n 个顶点，没有边的非连通图 T , 每个顶点自成一个连通分量

2. 在 E 中选最小权值的边,若该边的两个顶点落在不同的连通分量上，则加入 T 中；否则舍去，重新选择

3. 重复下去，直到所有顶点在同一连通分量上为止

4-14 假设依据Prim算法产生无向网的最小生成树，出发顶点为a，则被选择的顶点序列是( D )。



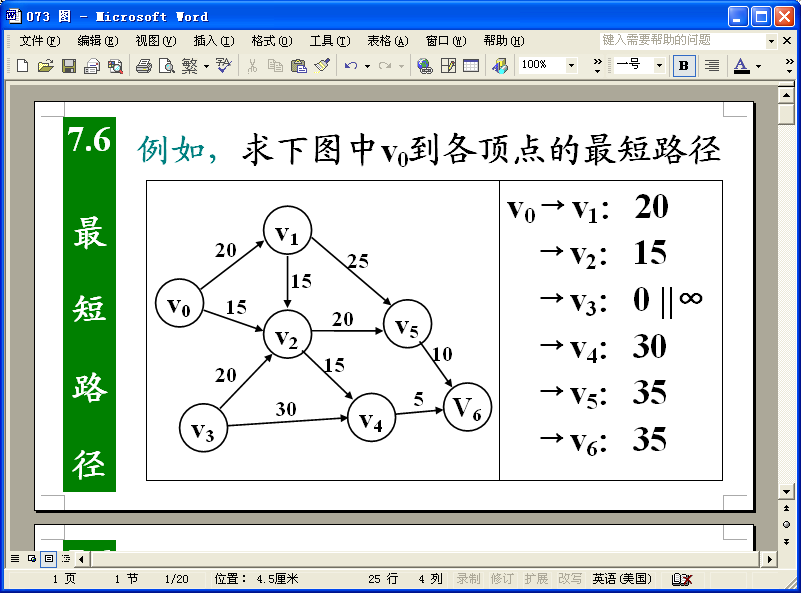
(A) a→b→c→d→e→f→g

(B) a→b→e→g→c→d→f

(C) a→e→d→b→c→f→g

(D) a→e→b→c→d→g→f

4-15 在有向图中，路径( C )是从v0出发的一条最短路径。



(A) v0→v1→v5

(B) v0→v2→v3

(C) v0→v2→v4

(D) v0→v2→v5→v6

4-16 采用邻接表存储结构，设计一个算法，判别无向图G中指定的两个顶点之间是否存在一条长度为k的简单路径。

注：简单路径是指顶点序列中不含有重复的顶点。

int exist\_path\_len(ALGraph G,int i,int j,int k)

{

if(i==j&&k==0) return true;

else if(k>0)

{

visited[i]=1;

for(p=G.vertices[i].firstarc;p!=NULL;p=p->nextarc)

{

m=p->adjvex;

if(!visited[m])

if(exist\_path\_len(G,m,j,k-1)) return true;

}//for

visited[i]=0;

}

return false;

}

4-17 设带权有向图G =(V, E)含有n个顶点、e条边，采用邻接矩阵Graph[n][n]作为存储结构。试设计算法Dijkstra(int V0，int n)，用于计算从源点V0到其它各顶点的最短路径。

void ShortestPath\_DIJ(AMGraph G, int v0){

//用Dijkstra算法求有向网G的v0顶点到其余顶点的最短路径

n=G.vexnum; //n为G中顶点的个数

for(v = 0; v<n; ++v){ //n个顶点依次初始化

S[v] = false; //S初始为空集

D[v] = G.arcs[v0][v]; //将v0到各个终点的最短路径长度初始化

if(D[v]< MaxInt) Path [v]=v0; //v0和v之间有弧，将v的前驱置为v0

else Path [v]=-1; //如果v0和v之间无弧，则将v的前驱置为-1

}//for

S[v0]=true; //将v0加入S

D[v0]=0; //源点到源点的距离为0

/\*―开始主循环，每次求得v0到某个顶点v的最短路径，将v加到S集―\*/

for(i=1;i<n; ++i){ //对其余n−1个顶点，依次进行计算

min= MaxInt;

for(w=0;w<n; ++w)

if(!S[w]&&D[w]<min)

{v=w; min=D[w];} //选择一条当前的最短路径，终点为v

S[v]=true; //将v加入S

for(w=0;w<n; ++w) //更新从v0出发到集合V−S上所有顶点的最短路径长度

if(!S[w]&&(D[v]+G.arcs[v][w]<D[w])){

D[w]=D[v]+G.arcs[v][w]; //更新D[w]

Path [w]=v; //更改w的前驱为v

}//if

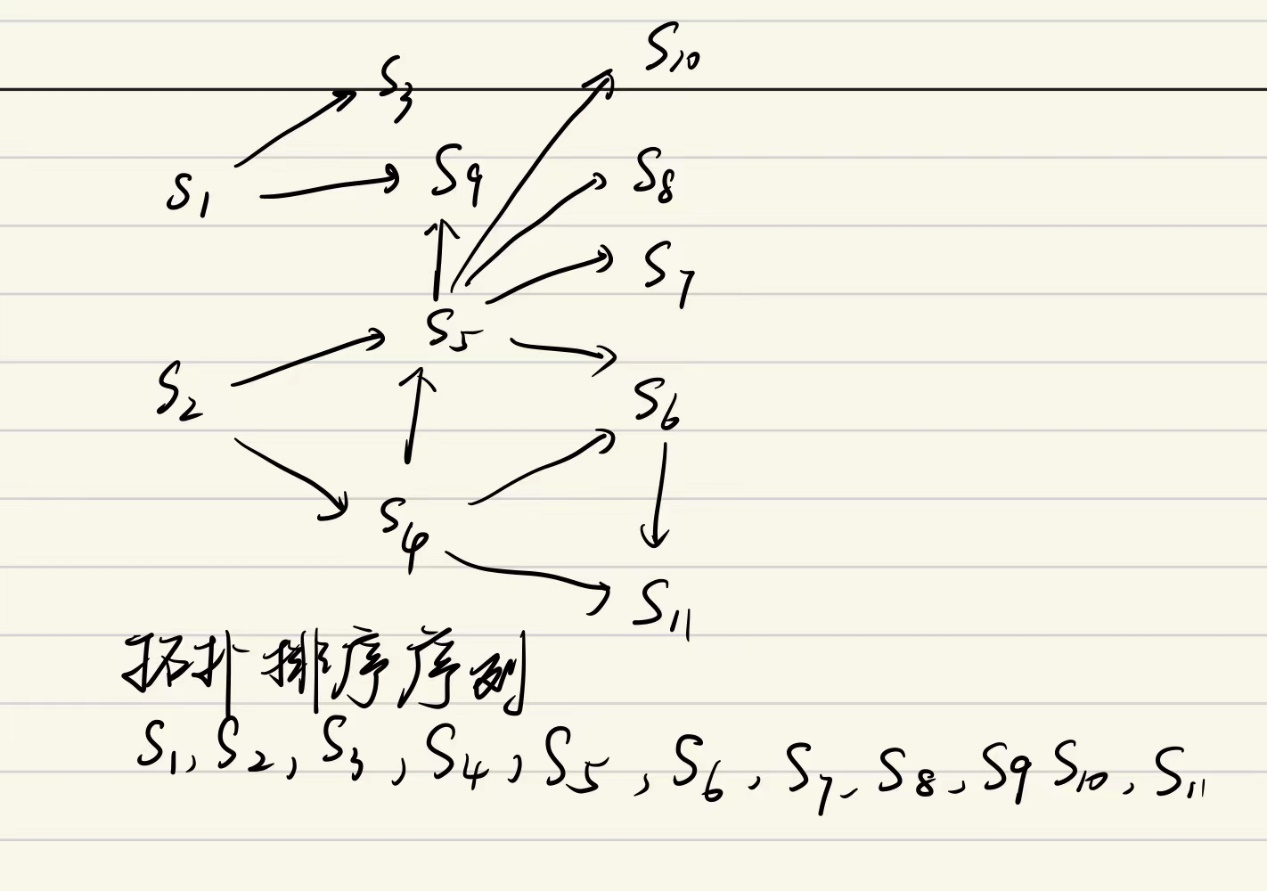
}//for

}

4-18 设软件工程专业开设的主要课程如表所示：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 代码 | 课程名称 | 先修课程 |  | 代码 | 课程名称 | 先修课程 |
| S1 | 高等数学 | 无 |  | S7 | 数据库系统 | S5 |
| S2 | 程序设计基础 | 无 |  | S8 | 编译技术 | S5 |
| S3 | 离散数学 | S1 |  | S9 | 算法分析 | S1, S5 |
| S4 | 计算机组成原理 | S2 |  | S10 | 软件工程导论 | S5 |
| S5 | 数据结构与算法 | S2, S3 |  | S11 | 计算机网络 | S4, S6 |
| S6 | 操作系统 | S4, S5 |  |  |  |  |

试根据先修课程要求绘制课程体系拓扑结构图(结点用课程代码表示)。



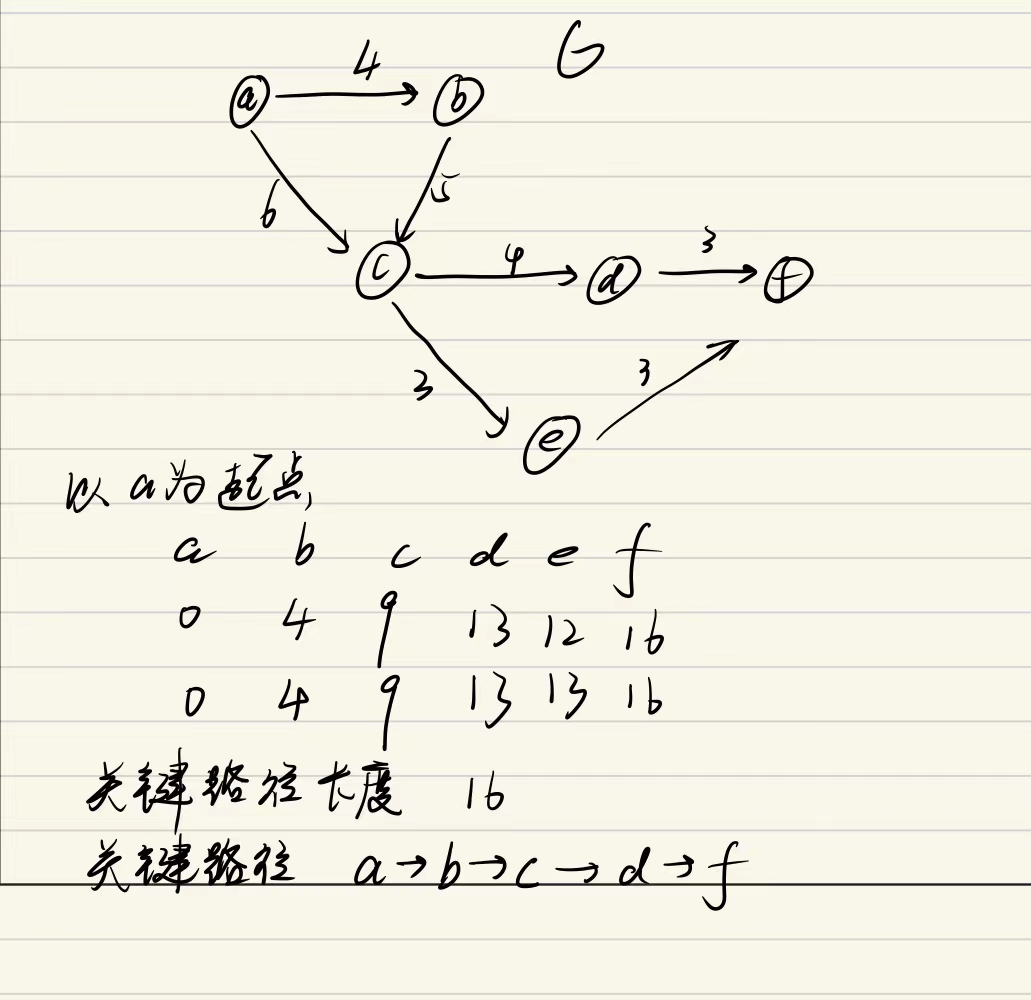
4-19 设含有6个顶点a, b, c, d, e, f的有向带权图G，其邻接矩阵如下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ＼ | a | b | c | d | e | f |
| a | 0 | 4 | 6 | 0 | 0 | 0 |
| b | 0 | 0 | 5 | 0 | 0 | 0 |
| c | 0 | 0 | 0 | 4 | 3 | 0 |
| d | 0 | 0 | 0 | 0 | 0 | 3 |
| e | 0 | 0 | 0 | 0 | 0 | 3 |
| f | 0 | 0 | 0 | 0 | 0 | 0 |

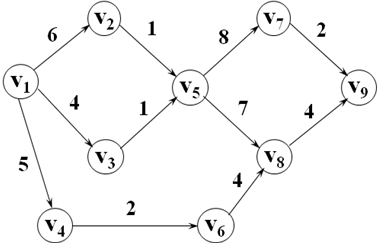
要求：

(1)画出有向带权图G；

(2)求图G的关键路径，并计算关键路径长度。



4-20 设v1是源点、v9是汇点，则在有向图中，( C )是一条关键路径。



(A) v1→v4→v6→v8→v9

(B) v1→v3→v5→v7→v9

(C) v1→v2→v5→v8→v9

(D) v1→v2→v5→v7→v9