

期末复习练习题2

开始时间 2023/12/18 14:44:00

结束时间 2024/01/13 16:44:00

答题时长 37560分钟

答卷类型 标准答案

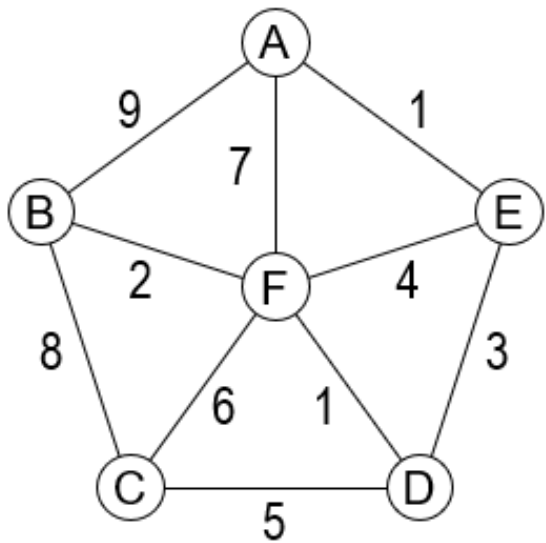
总分 50

填空题

得分：暂无 总分：10

4-1 在有n个顶点的有向图中，若要使任意两点间可以互相到达，则至少需要 (5分)条弧

4-2 请写出下图用普里姆算法从顶点A出发生成最小生成树每一步加入的边。



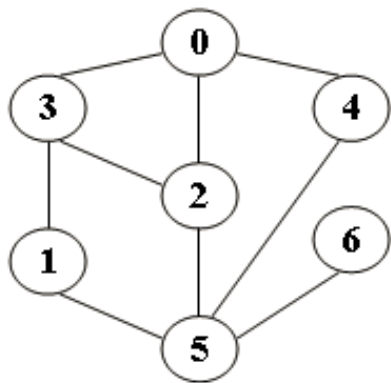
- (1) (1分)
- (2) (1分)
- (3) (1分)
- (4) (1分)
- (5) (1分)

注：顶点 X 到 Y 的无向边简记作：XY 或 YX。

程序填空题

得分：暂无 总分：10

5-1 的值本题要求建立一个无向图，采用邻接表做为存储结构。
例如



输入信息为：第一行给出图的顶点数n和边数e。第二行给出n个字符，表示n个顶点的数据元素的值。后面是e行，给出每一条边的两个顶点编号。

输出每个顶点的值以及各顶点的邻接点的值。

输入样例为：

7 9

0123456

0 2

0 3

0 4

1 3

1 5

2 3

2 5

4 5

5 6

输出样例为

0: 4 3 2

1: 5 3

2: 5 3 0

3: 2 1 0

4: 5 0

5: 6 4 2 1

6: 5

```
#include <stdio.h>
#include <stdlib.h>
#define MVNum 100 //最大顶点数
typedef struct ArcNode{ //表结点
    int adjvex; //邻接点的位置
    struct ArcNode * nextarc; //指向下一个表结点的指针
}ArcNode;
typedef struct VNode{
    char data; //顶点信息
    ArcNode * firstarc; //指向第一个表结点的指针
}VNode, AdjList[MVNum]; //AdjList表示邻接表类型
typedef struct{
    AdjList vertices; //头结点数组
    int vexnum, arcnum; //图的当前顶点数和边数
}ALGraph;
void CreatMGraph(ALGraph *G);/* 创建图 */
void printGraph(ALGraph G);/*输出图 */
int main()
{
    ALGraph G;
    CreatMGraph(&G);
    printGraph(G);
    return 0;
}
void CreatMGraph(ALGraph *G)
```

```

{
    int i,j,k;
    ArcNode *s;
    scanf("%d%d",&G->vexnum,&G->arcnum);
    getchar();
    for(i=0;i<G->vexnum;i++)
        scanf("%c",&G->vertices[i].data (2分));
    for(i=0;i<G->vexnum;i++)
        G->vertices[i].firstarc=NULL (2分);
    for(k=0;k<G->arcnum;k++) {
        scanf("%d%d",&i,&j);
        s=(ArcNode*)malloc(sizeof(ArcNode));
        s->adjvex=j;
        s->nextarc=G->vertices[i].firstarc;
        G->vertices[i].firstarc=s (2分);
        s=(ArcNode*)malloc(sizeof(ArcNode));
        s->adjvex=i;
        s->nextarc=G->vertices[j].firstarc (2分);
        G->vertices[j].firstarc=s (2分);
    }
}

void printGraph(ALGraph G)
{
    int i,j;
    ArcNode *p;
    for(i=0;i<G.vexnum;i++)
    {
        printf("%c:",G.vertices[i].data);
        for(p=G.vertices[i].firstarc;p;p=p->nextarc)
            printf(" %c",G.vertices[p->adjvex].data);
        printf("\n");
    }
}

```

编程题

得分：暂无 总分：30

7-1 有向图的拓扑序列 (20分)

输出有向图的拓扑序列。

输入格式:

输入第一行给出两个正整数，分别表示图的节点数 N ($1 < N \leq 10$)、边数 M (≤ 50)。随后的 M 行对应 M 条边，每行给出一对正整数，分别是有向边直接连通的两个节点的编号（编号从1到 N 依次进行）。

输出格式:

输出此图的拓扑序列，用一个空格隔开，最后也有一个空格；如果为非连通图或图中有回路，则在结尾处另起一行输出一个0。

由于拓扑序列是不唯一的，为了使得输出具有唯一的结果，我们约定以表头插入法构造邻接表，并且保证初始入度为0的节点仅有一个。当运行过程中同时出现多个入度为0的结点时，采用栈来保存。

输入样例1:

```
4 3
1 4
4 2
2 3
```

输出样例1:

```
1 4 2 3
```

输入样例2:

```
4 4
1 4
4 2
2 3
3 4
```

输出样例2:

```
1
0
```

7-2 双十一（10分）

双十一期间，某著名电商平台“东东”为应对销售高峰，准备在 n 个城市中再增加一个自营仓库，其要求是该仓库设在 n 个城市中的某个城市，且距离其他所有城市的最短距离之和最小。请编写程序帮助“东东”找出设立仓库的地点。假定 n 个城市编号为 0 至 $n-1$ ，它们之间至少有一个城市与其他所有城市可及。

输入格式:

输入包含多组数据。每组数据第一行为两个正整数 n 和 e ，均不超过 100 。 n 表示城市数。接下来 e 行表示两个城市间的距离信息，每行为3个非负整数 a 、 b 、 c ，其中 a 和 b 表示两个城市编号， c 表示城市间的距离。

提示：可使用EOF判断输入结束。

输出格式:

输出为一个整数，表示建立仓库的城市编号，如多个城市满足要求，则输出编号最小者。

输入样例:

```
6 5
0 1 1
0 2 1
0 3 1
0 4 1
0 5 1
4 5
0 1 1
0 2 5
1 2 2
1 3 4
```

2 3 1

输出样例:

0
1