

实验 6——图问题的实现

1634224 王显森

目录

一、实验目的

二、实验环境

1. 编程语言和开发环境

三、分析与设计

1. 需求分析

2. 类的设计

3. 设计细节

四、代码

五、实验结果

一、实验目的

设计一个能够产生简单背包问题所有解的小程序。

二、实验环境

MinGW-32;

2.1 编程语言和开发工具

编程语言: c++;

三、分析与设计

3.1 需求分析

设计一个程序，允许用户通过邻接链表或邻接矩阵创建和储存图，并且能够实现深度遍历和广度遍历，以及 Dijkstra 算法寻求选定定点到每个定点的最短距离。

四、代码

```
struct edge{//边的结构
    int dest;
    int weight;
    edge* next;
    edge(){
        dest=-1;
        weight=0;
        next=NULL;
    }
    edge(int v,int w){
        dest=v;
        weight=w;
        next=NULL;
    }
};
struct vertex{//节点的结构
    int ver;
    string name;
    edge* head;
    vertex(){
        ver=-1;
        head=NULL;
    }
};
```

```
bool insert_vertex(int n, string value){//点插入
    if(n>=size){
        cout<<"the index is wrong"<<endl;
        return false;
    }
    if(node[n].ver==-1){
        node[n].ver=n;
        node[n].name=value;
        return true;
    }
    else{
        return false;
    }
}
```

```

bool insert_edge(int v1, int v2,int w){//边插入
    if(node[v1].ver==-1){
        cout<<"vertex "<<v1<<" is not exsisted."<<endl;
        return false;
    }
    if(node[v2].ver==-1){
        cout<<"vertex "<<v2<<" is not exsisted."<<endl;
        return false;
    }
    edge* temp=node[v1].head;
    edge* temp1=node[v2].head;
    if(temp==NULL){
        temp= new edge(v2,w);
        node[v1].head=temp;
    }
    else{
        while(temp->next!=NULL){
            if(temp->dest==v2){
                cout<<"this edge has been exsisted"<<endl;
                return false;
            }
            temp=temp->next;
        }
        temp->next=new edge(v2,w);
    }
    if(temp1==NULL){
        temp1= new edge(v1,w);
        node[v2].head=temp1;
    }
    else{
        while(temp1->next!=NULL){
            if(temp1->dest==v1){
                cout<<"this edge has been exsisted"<<endl;
                return false;
            }
            temp1=temp1->next;
        }
        temp1->next=new edge(v1,w);
    }
    return true;
}

```

五、实验结果

构造

```
命令提示符 - a.exe
C:\Users\lenovo\Desktop>a.exe
Construct graph:

Please input the number of vertex:
7
Please input the name of vertex(order[1~n]):
bj
xa
zz
xz
cd
gz
sh
Please input the edge("-1 -1 -1" to end):
1 2 2553
1 3 695
1 4 704
2 3 511
2 5 811
3 4 359
3 6 1579
4 7 651
5 6 2368
6 7 1385
-1 -1 -1
```

界面

```
命令提示符 - a.exe
Graph: 1. Traverse 2. Dijkstra 0. Quit
demand$: ~~~~
_
```

遍历

```
命令提示符 - a.exe
Graph: 1. Traverse 2. Dijkstra 0. Quit
demand$:~~~ 1
Please input begin vertex:
1
DFS:
1(bj) ->2(xa) ->3(zz) ->4(xz) ->7(sh) ->6(gz) ->5(cd) ->
BFS:
1(bj) ->2(xa) ->3(zz) ->4(xz) ->5(cd) ->6(gz) ->7(sh) ->
请按任意键继续. . .
```

Dijkstra:

```
命令提示符 - a.exe
Graph: 1. Traverse 2. Dijkstra 0. Quit
demand$:~~~ 2
Please input begin vertex:
1
Dijkstra:
1: 1(bj)->3(zz) weight: 695
2: 3(zz)->4(xz) weight: 359
3: 3(zz)->2(xa) weight: 511
4: 4(xz)->7(sh) weight: 651
5: 2(xa)->5(cd) weight: 811
6: 7(sh)->6(gz) weight: 1385
请按任意键继续. . .
```

PS:本程序仅实现了在临接链表下的存储、DSF、BSF、Dijkstra 算法，但源代码中包含临接矩阵的类实现。