

10.图和树基础

1、邻接矩阵的实现

```
1  #include <iostream>
2  #include <cstring>
3
4  using namespace std;
5  int main()
6  {
7      int G[6][6];
8      memset(G,0,sizeof(G));
9      int m;
10     cin >> m;
11     for (int i = 0; i < m ;i++)
12     {
13         int a,b;
14         cin >> a >> b;
15         G[a][b] = 1;
16     }
17     for (int i = 1; i <= 5; i ++ )
18     {
19         int sum = 0;
20         for (int j = 1; j <= 5; j ++ )
21         {
22             if (G[i][j] == 1 && G[j][i] == 1)
23             {
24                 sum++;
25             }
26         }
27         cout << i << "有" << sum <<"个真正的朋友"<<endl;
28     }
29     return 0;
30 }
```

2、邻接表的实现

```
1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5  int main()
6  {
7      vector<int> G[11];
```

```

8     int m;
9     cin >> m;
10    for (int i = 0; i < m; i++)
11    {
12        int a,b;
13        cin >> a >> b;
14        G[a].push_back(b);
15        G[b].push_back(a);
16    }
17    for(int i = 1; i <= 10; i++)
18    {
19        cout << i << ":";
20        for (int j = 0; j < G[i].size();j++)
21        {
22
23            cout << G[i][j] << " ";
24        }
25        cout <<endl;
26    }
27
28    return 0;
29 }

```

3、带权值的图

```

1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  struct node
7  {
8      int v,w;
9  };
10 vector<node> G[11];
11 void insert1(int u, int v, int w)
12 {
13     node temp;
14     temp.v = v;
15     temp.w = w;
16     G[u].push_back(temp);
17 }
18 void insert2(int u, int v, int w)
19 {
20     insert1(u,v,w);
21     insert1(v,u,w);
22 }
23
24 int main()
25 {
26     int m;

```

```

27     cin >> m;
28     for (int i = 0; i < m; i++)
29     {
30         int u,v,w;
31         cin >> u >> v >> w;
32         insert2(u,v,w);
33     }
34     for(int i = 1; i <= 10; i++)
35     {
36
37         for (int j = 0; j < G[i].size();j++)
38         {
39             cout << "("<< i << "," << G[i][j].v << "," <<G[i][j].w <<")" <<
40         }
41     }
42
43     return 0;
44 }

```

4、关系查询

```

1  #include <iostream>
2  #include <map>
3
4  using namespace std;
5  int G[500][500];
6  map<string,int> dict;
7  int ids;
8  int find(string a)
9  {
10     if(dict.find(a) == dict.end())
11     {
12         dict[a] = ++ ids;
13     }
14     else
15     {
16         return dict[a];
17     }
18 }
19
20 int main()
21 {
22     memset(G,0,sizeof(G));
23     int n;
24     for (int i = 0; i < n; i ++)
25     {
26         string a, b;
27         cin >> a >> b;
28         int x = find(a);
29         int y = find(b);
30         G[x][y] = G[y][x] = 1;

```

```
31     }  
32     return 0;  
33 }
```