Readme

Network.h

实现了无向图的临接表的基本操作、图属性的计算和经典网络模型的生成。

约定

名称	约定
n	节点数
Adj_List	邻接表
u	节点u
v	节点v
k	平均度
p	概率

功能

邻接表操作

1.邻接表的初始化

```
vector< vector<int> > Init(int n);
```

2.邻接表的撤销

```
vector< vector<int> > Destroy(vector<vector<int> > Adj_List)
```

3.搜索边(u,v)

```
int Exist(vector<vector<int> > Adj_List,int n,int u, int v)
```

4.插入边(u,v)

```
vector< vector<int> > Insert(vector< vector<int> > Adj_List,int n,int u, int v)
```

5.删除边(u,v)

```
vector< vector<int> > Remove(vector< vector<int> > Adj_List, int n, int u, int v)
```

6.输出邻接表（普通格式）

```
void Raw_printAdjList(vector< vector<int> > Adj_List)
```

7.输出邻接表(https://csacademy.com/app/graph_editor/的格式)

```
void Csa_printAdjList(vector< vector<int> > Adj_List)
```

图的属性计算

1.计算各个点的聚类系数

```
vector<double> Cal_Ck (vector<vector<int> > Adj_List,int n)
```

2.统计最大度

```
int MaxDegree (vector<vector<int> > Adj_List,int n)
```

3.统计各个节点的度

```
vector<int> CountDegree (vector<vector<int> > Adj_List,int n )
```

4.度分布

```
vector<double> DegreeDistribution(vector<int> Degree, int n)
```

5.累积度分布

```
vector<double> CumulaDDistribution(vector<int> Degree, int n)
```

经典网络模型

1.生成ER图

```
vector<vector<int> > CreateERGraph(vector<vector<int> > Adj_List,int n,int k)
```

2.生成最近邻耦合网络

```
vector< vector<int> > CreateNCCNetwork(vector< vector<int> > Adj_List,int n,int k)
```

3.生成WS小世界网络（随机重连）

```
vector< vector<int> > CreateWSNetwork(vector< vector<int> > Adj_List,int n, int
k,double p)
```

4.生成完全图

```
vector< vector<int> > CreateCGraph(vector< vector<int> > Adj_List,int n)
```

5.生成BA无标度网络（优先连接）

```
vector< vector<int> > CreateBANetwork(vector< vector<int> > Adj_List, int n,int m0,
int t, int m)//m0：完全图的节点数，t：总共增加t个节点，m：新增节点每次增加m条边
```

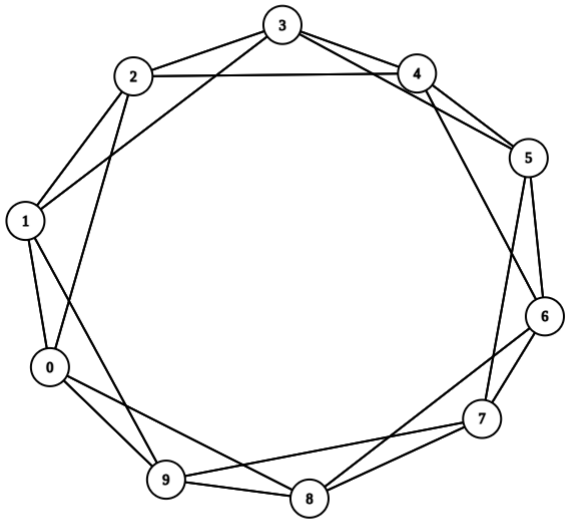
使用例

例1：生成一个WS小世界网络

```
#include <cstdlib>
#include <ctime>
#include <iostream>
#include <vector>
#include <algorithm>
#include<time.h>
#include <stdio.h>
#include"network.h"
using namespace std;
  int main(){
    vector<vector<int> > WSGraph;
    int n=10;
    int k=4;
    double p=0.2;
    WSGraph=Init(n);
    WSGraph=CreateNCCNetwork(WSGraph,n,k);
    WSGraph=CreateWSNetwork(WSGraph,n,k,p);
    Csa_printAdjList(WSGraph);
    return 0;
  }
```

结果图：

最近邻耦合网络



WS小世界网络