

欧拉路径或欧拉回路的一种算法

文/孙风庆

摘要

本文主要是,给出求欧拉路径或欧拉回路的一种算法,该算法能够找出从某顶点出发的所有欧拉路径或欧拉回路,通过图的邻接矩阵存储方式和C++语言实现。

【关键词】图 邻接矩阵 编号 顶点的度 欧拉路径 欧拉回路

1 定义图(邻接矩阵)类

```
template<class T>
```

```
class Graph
{ private:
    T *ver ; // 数组 ver 存储图中各个顶点的
    数据
    int **edge ; // 数组 edge 为图的邻接矩阵
    int vers ,edges ; // 变量 vers 和 edges 分别
    存储图的顶点个数和边个数
    bool tag ; //tag 是个标记变量,若值为 0
    表示是无向图,否则是有向图
    bool *visit ; // 数组 visit[k] 用于标记编号
    为 k 的边是否已被访问
    int *path ,n ,m ; // 数组 path 用于暂存所
```

```
找出的当前路径,变量 n 暂存该当前路径中的
顶点个数,变量 m 用于存储找到的所有路径的
条数
    int deg(int v) ; // 求顶点 v 的度
    void find(int v) ; // 查找从顶点 v 出发的所
    有欧拉路径或欧拉回路
    public:
        Graph(bool p ,T a[] ,int n ,int e) ; // 构造
        函数
        void euler(int v) ; // 求从顶点 v 出发的
        所有欧拉路径或欧拉回路
    } ;
```

<< 上接 195 页

电子电气产品中 6 种有害物质的检测方法有很多,如原子光谱法、分光光度法、变温红外光谱分析法、X 射线荧光分析法、电位溶出法等多种方法。常用的仪器设备有电感耦合等离子体原子发射光谱仪、气相色谱-质谱联用仪、原子吸收分光光度计和高效液相色谱仪等。下面就这些方法中的一些常用方法进行介绍。

3.3.1 电感耦合等离子体原子发射光谱法

原子发射光谱分析是利用元素在受热或电激发时,在从基态跃迁到激发态最后返回到基态的过程中,发射出特征光谱,然后根据特征光谱进行定性、定量分析的方法。由于该方法具有良好的选择性,所以在检测行业备受关注,广泛应用于各种分析测试中。此法对于铅、镉、汞、铬等重金属,ICP-AES 法是最常用也最实用的方法。

3.3.2 分光光度计比色法

分光光度计比色法主要是用于对六价铬进行准确测定。测定时,首先要对三价铬进行洗分离,而后把洗分离后的样品浸泡在 pH 溶液中,溶液里含有六价铬会与显色剂二苯基二氮发生反应因而呈现红色,最后用分光光度计测定吸光度,就能得到六价铬含量。

3.3.3 原子吸收光谱法

原子吸收光谱法主要是根据气态原子对同种原子发射出的特征光谱辐射会具有一定的吸收能力原理,通过测量吸光度就可进行定量检测。其方法一般包括石墨炉原子吸收光谱法、火焰原子吸收光谱法等。

由于汞的易挥发性,原子吸收光谱法是目前对汞元素最常用的检测方法,我国许多标准中,对汞元素的测定也基本都是采用该方法。同时原子吸收光谱法也可以对铅、镉、铬等其他种类的重金属进行有效检测。

3.3.4 原气相色谱-质谱法

气相色谱-质谱联用是由气相色谱仪与质谱仪两种设备经由一定的接口技术联用而成的,是仪器联用技术的典型代表作。目前,许多实验室也都是把气相色谱-质谱联用作为最主要的定性确认手段,同时该方法也是分析复杂混合物最为有效的手段之一。电子电气产品中 6 类有毒有害物质中的多溴联苯醚和多溴联苯的含量测定都是用该方法进行的。

4 总结

随着电子电气产品的更新换代,其废弃物的污染也日益严重,各种毒害物质严重威胁

人类健康。因此备受关注,相关标准也越来越严格。在此本文简单介绍了一下电子电气产品中有毒有害物质的危害及相关检测方法,希望能对相关生产企业和检测机构带来帮助。

参考文献

- [1] 范岑亮. 电子电气产品中六类有害物质含量的检测方法[J]. 环境技术, 2006(03).
- [2] 何重辉, 曹蓉. 电子电气产品中限用物质的国际标准化研究[J]. 中国测试, 2010(01).
- [3] 唐式金. 电子电气产品 RoHS 认证简介——限制在电子电气产品中使用某些有害物质[J]. 江苏电器, 2007(03).
- [4] 闫军, 高峰, 刘霁欣. 蒸汽发生-原子荧光法测定电子电气产品中的 Cr(VI)[J]. 现代仪器, 2006(02).
- [5] 陈泽勇, 张进军. IEC62321 有害物质检测方法进展[J]. 信息技术与标准化, 2009(07).

作者单位

西安市产品质量监督检验院 陕西省西安市 710065

(1) 构造函数 $\text{Graph}(p, a, n, e)$ 是通过数组 a 中的数据创建一个有 n 个顶点和 e 条边的图, 若 p 值为 0, 则建立无向图, 否则建立有向图; 另外, 为了求欧拉路径, 给图中的每一条边进行编号 (从 0 开始), 并用此编号作为相应边的权值。

```
template<class T>
Graph<T>::Graph(bool p, T a[], int n, int e)
{ tag=p; vers=n; edges=e;
  int i, j, k;
  // 初始化
  ver=new T[vers]; for(i=0; i<vers; i++)
ver[i]=a[i];
  edge=new int*[vers];
  for(i=0; i<vers; i++){edge[i]=new
int[vers]; for(j=0; j<vers; j++)edge[i][j]=-1; }
  // 建立图中的各个边, 并给边进行编号
  cout<< " 依次分别输入图中的每条边所依
附的两个顶点: " <<endl;
  for(k=0; k<edges; k++){cin>>i>>j;
edge[i][j]=k; if(tag==0)edge[j][i]=k; }
  return ;
}
```

(2) 函数 $\text{deg}(v)$ 是求顶点 v 的度

```
template<class T>
int Graph<T>::deg(int v)
{ int d=0, i;
  for(i=0; i<vers; i++)if(edge[v][i]!=-1)
d++;
  if(tag!=0){for(i=0; i<vers; i++)if(edge[i]
[v]!=-1)d++; }
  return d;
}
```

2 寻找欧拉路径或欧拉回路

算法的基本思想是: 按照深度优先搜索的方法, 查找从某个顶点出发的所有欧拉路径或欧拉回路。由欧拉路径的定义可知, 欧拉路径或欧拉回路的长度恰好等于图的边数。

(1) 函数 $\text{find}(v)$ 是查找图中从顶点 v 出发的所有欧拉路径或欧拉回路。

```
template<class T>
void Graph<T>::find(int v)
```

```
{ path[n++]=v; // 先将顶点 v 添加到当前
路径 path 中
  int i, j, k;
  if( n==edges+1 )// 找到一条从顶点 v 出发
的欧拉路径或欧拉回路
  { for(i=0; i<n; i++){j=path[i];
cout<<ver[j]<<" , " ; }// 输出欧拉路径或欧拉
回路
    cout<<endl;
    m++;
  }
  else for(i=0; i<vers; i++)
  { k=edge[v][i]; // 取出以顶点 v 为首的边
的编号
    if(k!=-1&&visit[k]==0){ visit[k]=1;
find(i); }
    n--; // 再从当前路径 path 中删除顶点 v,
并将以顶点 v 为尾的边的访问标记重新置为 0
    if(n>0){ k=edge[path[n-1]][v];
visit[k]=0; }
    return ;
  }
}
```

(2) 函数 $\text{euler}(v)$ 是求图中从顶点 v 出发的所有欧拉路径或欧拉回路。

```
template<class T>
void Graph<T>::euler(int v)
{ if(v<0||v>=vers) return ;
  // 先判断图中是否存在欧拉路径或欧拉
回路
  int i, d, p, u[2];
  p=0; // 统计图中奇数度顶点的个数 p
  for(i=0; i<vers; i++){ d=deg(i);
if(d%2==1)u[p++]=i; }
  if(p!=0&&p!=2){cout<< " 图中不存在欧
拉路径!" <<endl; return ; }
  // 开始求欧拉路径或欧拉回路
  path=new int[edges+1]; visit=new
bool[edges];
  for(i=0; i<edges; i++)visit[i]=0; n=0;
m=0; // 初始化
  if(p==0)// 图中存在欧拉回路
  { cout<< " 从顶点 " <<ver[v]<< " 出发
```

```
的欧拉回路: " <<endl;
    find(v);
    if(m==0)
      cout<< " 从顶点 " <<ver[v]<< " 出发
没有欧拉路径或欧拉回路. " <<endl;
    else
      cout<< " 共有 " <<m<< " 条欧拉回路
. " <<endl;
  }
  else // 图中存在欧拉路径
  { int w; cout<< " 图中的欧拉路
径: " <<endl;
    w=u[0]; find(w);
    if(m==0){for(i=0; i<edges; i++)
visit[i]=0; n=0; w=u[1]; find(w); }
    if(m==0)cout<< " 图中没有欧拉路
径. " <<endl;
    else cout<< " 从顶点 " <<ver[w]<< "
出发有 " <<m<< " 条欧拉路径. " <<endl;
  }
  return ;
}
```

参考文献

- [1] 王红梅的数据结构 (C++ 版) [M]. 北京: 清华大学出版社, 2009(7).
- [2] 夏克俭的数据结构与算法 [M]. 北京: 国防工业出版社, 2000(7).
- [3] 严蔚敏的数据结构 (C 语言版) [M]. 北京: 清华大学出版社, 2003(12).
- [4] 翁惠玉的数据结构与实现 [M]. 北京: 高等教育出版社, 2009(8).

作者简介

孙风庆, 1963 年 12 月出生, 男, 山东省定陶县人, 副教授, 大学学历, 工学学士, 主要研究方向是计算方法和数据结构。

作者单位

山东德州职业技术学院计算机系 山东省德州市 253034