《集合与图论》的联合实验课程——以欧拉图构造性证明的算法实现为例*

吴永辉

上海智能信息处理重点实验室，复旦大学计算机科学技术学院

上海，200433

摘　要　《集合与图论》是计算机科学与技术的理论基础课。《集合与图论》与程序设计类课程的联合实验课程，基于程序设计竞赛试题，以提高学生综合应用知识编程解决问题的能力为目标。本文详述《集合与图论》和《数据结构》的联合实验课程“欧拉图构造性证明的算法实现”。实验由求解欧拉链的程序设计竞赛试题及其解析构成；学生完成基于 DFS 的欧拉图定理的构造性证明；体会证明过程就是发现欧拉链的算法；在编程解决问题的过程中综合运用 DFS、图的建模、并查集判断图的连通性等。

关键字　程序设计，集合与图论，欧拉图，DFS，构造性证明，并查集，图的建模

A Joint Experimental Curriculum for Set and Graph Theory: The Algorithm Implementation for the Constructive Proof of Euler Graph

Yonghui Wu

Shanghai Key Laboratory of Intelligent Information Processing, School of Computer Science
Fudan University
Shanghai 200433, China
yhwu@fudan.edu.cn

Abstract— “Set and Graph Theory” is the theoretical foundation for computer science and technology. We construct joint experimental curriculums for set and graph theory and programming curriculums based on programming contest problems, to improve students’ skills solving problems by programming and comprehensively applying knowledge. The paper specifies a joint experimental curriculum for set and graph theory and data structure “the algorithm implementation for the constructive proof of Euler graph”. A programming contest problem solving Eulerian circuit and its analysis constitute the experiment. Students are required to finish the constructive proof of Euler graph based on DFS, and realize the proof is the algorithm finding the Eulerian circuit. Solving the problem by programming requires students to comprehensively apply DFS, Union-Find Set used to determine graph connectivity, graph modeling, and so on.

Keywords—Programming, Set and Graph Theory, Euler Graph, DFS, Constructive Proof, Union-Find Set, Graph Modeling

1　引　言

《集合与图论》是计算机科学的理论基础课，包括集合论、图论、组合数学初步。其教学目的，一方面，奠定学生的计算机专业知识体系的理论基础，磨练学生的思维方式数学化；另一方面，展现《集合与图论》作为理论基础课程与其他课程的关系，使得同学们更好地理解计算机的知识体系结构，以及应用理论知识解决实际问题。

程序设计类课程和集合与图论的教学，已经经历了多年的课程建设，也已经有了很多经典的教材。然而，一方面，课程之间彼此相对独立；另一方面，就课程本身而言，目前的程序设计类课程的教学还是以课堂教学为主，无法系统、全面地磨练学生编程解决问题的能力；而集合与图论课程也没有很好地让学生体会到计算机理论基础与计算机主干课程之间的关联，也没有很好地磨练学生应用理论解决实际问题的能力。

程序设计竞赛是“编程解决问题”的竞赛，在 1980 年代中后期走向成熟之后，近 40 年来，累积了海量的

*基金资助：本文得到复旦大学本科教改项目（FD2022A106），教育部产学合作协同育人项目（CCS54WHZ4220532）支持。

试题。这些来自全球各地，凝聚了无数命题者的心血和智慧的试题，不仅可以用于程序设计竞赛选手的训练，而且可以用于教学和实验，系统、全面提高学生综合应用知识编程解决问题的能力。

为此，笔者对海量的程序设计竞赛的试题进行了分析和整理，从中精选出相关试题，基于程序设计竞赛的知识体系，以系统、全面地提高学生编程解决问题的能力为目标，以实验为单位，进行教材建设和课程建设。在教材建设方面，编写“大学程序设计课程与竞赛训练教材”系列，在中国大陆出版简体中文版，在台湾出版繁体版，并一直改进，脱胎换骨地出版新版；在美国由 CRC Press 出版英文版，全球发行[1][2][3][4][5][6][7][8]。在课程建设方面，教学方法采用以编程实验为基础的案例教学，并采用在线测试系统、虚拟在线竞赛等信息化技术；不仅建设程序设计类实验课程，并以实验教改为切入点，融入大学程序设计类课程中[9][10][11][12][13]；而且，建设《集合与图论》与程序设计类课程的联合实验，以编程解题为抓手，构建《集合与图论》和程序设计类课程的关联，将磨练学生的思维方式数学化和提高编程解决问题的能力相结合，系统提高学生综合应用知识编程解决问题的能力。

本文详述《集合与图论》和《数据结构》的联合实验课程“欧拉图构造性证明的算法实现”。首先，本文介绍并查集，以及通过并查集判断图的连通性；并查集的实验，也是集合与图论与数据结构的联合实验；同时，也是欧拉图构造性证明的算法实现实验的前导实验。然后，本文详述《集合与图论》和《数据结构》的联合实验课程“欧拉图构造性证明的算法实现”，由基于 DFS 的欧拉图构造性证明引出求解欧拉链的算法，由程序设计竞赛试题 The Necklace 及其解析给出编程解题所需要综合应用的知识：图的建模、DFS、并查集判断图的连通性。

2 并查集及图的连通性的判断

在现实中，存在“物以类聚，人以群分”的关系，定义如下：

定义2.1（划分，块）． 设S是任意一个集合。$S_i \subseteq S$，$S_i \neq \varnothing$，i=1,2, …, n。如果$S_1 \cup S_2 \cup \ldots\ldots \cup S_n = S$，并且$S_i \cap S_j = \varnothing$（$i, j$=1, 2, …, n，$i \neq j$），则称$\pi=\{S_1, S_2, \ldots, S_n\}$是$S$的一个**划分**，其中每个$S_i$称为划分$\pi$的一个**块**。

由于这类问题主要涉及对集合的合并和查找，因此也称$\pi=\{S_1, S_2, \ldots, S_n\}$为并查集。

并查集维护互不相交的集合$S_1, S_2, \ldots, S_n$，每个集合S_i都有一个特殊元素$rep[S_i]$，称为集合S_i的代表元。并查集支持如下三种操作：

（1） *make_set*(x)：加入一个含单元素的集合$\{x\}$到并查集$\pi=\{S_1, S_2, \ldots, S_n\}$中，则$rep[\{x\}]=x$。注意$x$不能被包含在任何一个$S_i$中，因为在$\pi$中任何两个集合都是不相交的。初始时，对每个元素$x$执行一次*make_set*($x$)。

（2） *join*(x, y)：把x和y所在的集合S_x和S_y合并，也就是说，从π中删除S_x和S_y，并加入$S_x \cup S_y$。

（3） *set_find*(x)：返回x所在集合S_x的代表元$rep[S_x]$。

并查集的存储结构一般采用树结构，每个集合用一棵树表示，集合中的每个元素表示为树中的一个节点，根为集合的代表元（如图1）。

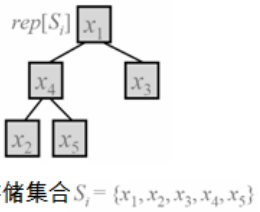

图 1

每个节点p设一个指针*set*[p]，记录它所在树的根节点序号。如果*set*[p]<0，则表明p为根节点。初始时，为每一个元素建立一个集合，即 *set*[x]=-1（$1 \le x \le n$）。

对于查找操作，我们采用边查找边“路径压缩”的办法，在查找过程的同时，也减少树的深度。例如在图2(a)所示的集合中，查找元素y_2所在集合的代表元，就从y_2出发，沿路径y_2-y_3-y_1-x_1查找到x_1，并依次将路径上的y_2、y_3、y_1的*set*指针设为x_1，如图2 (b)。

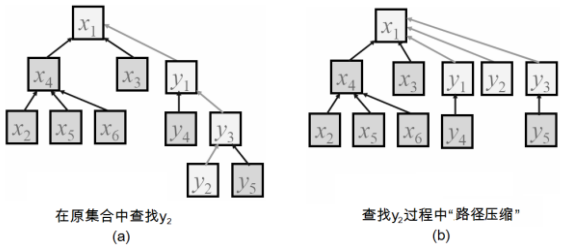

图 2

边查找边“路径压缩”的算法如下：

首先，从节点 x 出发，沿 *set* 指针查找节点 x 所在树的根节点 f（*set*[f]<0）。然后进行路径压缩，将 x 至 f 的路径上经过的每个节点的 *set* 指针都指向 f。查找过程如下：

```
int set_find(int p)
// 查找 p 所在集合的代表元，用路径压缩优化
{
    if (set[p]<0)
        return p;
    return set[p]=set_find(set[p]);
```

}

合并操作只需将两棵树的根节点相连即可。例如，将*x*所在的集合（树根*fx*）并入*y*所在的集合（树根*fy*），即以*fx*为根的子树上根节点的*set*指针指向*fy*，如图3。

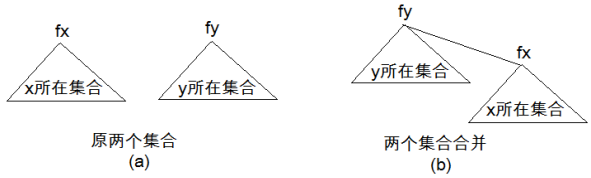

图 3

合并的算法如下：

计算*x*元素所在并查集的树根*fx*和*y*元素所在并查集的树根*fy*。如果*fx*==*fy*，则说明元素*x*和元素*y*在同一并查集中；否则将*x*所在的集合并入*y*所在的集合，也就是将*fx*的*set*指针设为*fy*：

```
void join(int p, int q)
// 将 p 所在的集合并入 q 所在的集合
{
    p=set_find(p);
    q=set_find(q);
    if (p!=q)
        set[p]=q;
}
```

“路径压缩”是在查找过程中进行，而不在合并过程中进行，因为合并需要查找待合并的两个集合的根节点，路径压缩实际上就是减少查找中树的深度，从而减少查找的时间消耗。

判断图的连通性，可以用 BFS、DFS、Warshell 算法等，也可以通过并查集判断图的连通性：根据图的边给出节点集合的划分。一条边关联的两个点是相连的，就被划分到同一个集合中。初始时，图中每个节点构成一个集合。然后，依次输入边；如果输入的边所关联的两个节点是在同一个集合中，则这两个节点已经是连通的；如果输入的边所关联的两个节点是在两个不同集合中，则这两个集合中的节点至少可以通过该边连通，就将这两个集合进行合并。重复以上过程，就可得到节点集合的划分。如果所有节点在一个集合中，则图是连通图；否则，图不连通，而节点集合的每个划分是一个连通分支。

3 欧拉图构造性证明的算法实现实验

3．1 基于 DFS 的欧拉图构造性证明

构造性证明，就是定理说明存在一种特定的对象，而证明方法就是说明如何构造这样的对象。

定义3.1（欧拉图）． 如果在图*G*中具有一条包含*G*中所有边的闭链，则称它为欧拉闭链，简称为欧拉链，称*G*为欧拉图。

定理 3.1 *G* 是连通图，则 *G* 是欧拉图当且仅当 *G* 的所有节点都是偶节点。

证明：如果图 *G* 是欧拉图，则在 *G* 中有一条包含 *G* 中所有边的闭链（欧拉链）$x_1\ x_2\ \dots\ x_m$，且 $x_1= x_m$。如 x_i 在序列 $x_1\ x_2\ \dots\ x_m$ 中出现 k 次，$1\le i\le m-1$，则 $d(x_i)=2k$。所以 *G* 的所有节点都是偶节点。

图 *G* 是连通图且每个节点都是偶节点，可以用 DFS 在图 *G* 中搜索出一条闭链 *C*。如果 *C* 不是欧拉链，则在 *C* 中必有一个节点 v_k，其度数大于在 *C* 中 v_k 连接的边的数目，就用 DFS 从 v_k 开始搜索一条边不在 *C* 中的闭链 *C*’。如果 *C*∪*C*’=*G*，则 *C*∪*C*’是欧拉链；否则同理，在 *C*∪*C*’中必有一个节点 v_k’，其度数大于在 *C*∪*C*’中 v_k’连接的边的数目，再用 DFS 从 v_k’开始搜索一条边不在 *C*∪*C*’中的闭链 *C*”，加入到 *C*∪*C*’中；以此类推，直到获得欧拉链。

显然，必要性的证明过程也是获得欧拉链的算法。

3．2 欧拉图构造性证明的算法实现的实验：The Necklace

由学生编程解决程序设计竞赛试题“The Necklace”。所完成的程序要在规定的时间和内存限制下，通过官方测试数据，或通过在线测试。

程序设计竞赛试题“The Necklace”描述如下：

我妹妹有一条用彩色珠子做的漂亮项链。每个珠子由两种颜色组成，相继的两个珠子在邻接处共享一种颜色，如图4所示：

图 4

有一天，项链线断了，珠子撒了一地。妹妹收集了散落在地上的珠子，但无法肯定是否收齐。她来找我帮忙，想知道用目前收集的珠子是否能够联成项链。

请您帮我写一个程序解决这个问题。

输入

输入包含*T* 个测试用例，输入的第一行给出整数*T*。

每个测试用例的第一行给出一个整数*N* (5≤*N*≤100)，表示我妹妹收集到的珠子的数目。接下来的*N* 行每行包含两个整数，表示一个珠子的两种颜色，颜色用从1到50的整数表示。

输出

对于入中的每个测试用例，首先输出测试用例编号，如样例输出；如果无法做出项链，输出一行“some beads may be lost”；否则，输出*N*行，每行用珠子两端的颜色对应的两个整数描述一颗珠子，在第*i*行第2个整数要和第*i* + 1行的第1个整数相同。此外，在第*N* 行的第2个整数要和第1行的第1个整数相等。可能存在多解，任何一个解都是可接受的。

在两个连续的测试用例之间输出一个空行。

样例输入	样例输出
2 5 1 2 2 3 3 4 4 5 5 6 5 2 1 2 2 3 4 3 1 2 4	Case #1 some beads may be lost Case #2 2 1 1 3 3 4 4 2 2 2

在线测试：UVA 10054，UVA 2036

时限: 1000MS，内存限制：65536 K

3．3 实验原理和方案：算法+数据结构=程序

图提供了一个自然的结构，由此产生的数学模型几乎适合于所有科学（自然科学和社会科学）领域，只要这个领域研究的主题是“对象”与“对象”之间的关系。

建立图模型的首要问题是：在图中什么为节点（对象）？什么为边（对象之间的关系）？

对于试题“The Necklace”，如果以直观的做法，珠子为节点，珠子串接的可能为边，则建立的图无法将问题的内容表达完全。如3颗珠子（red | green），（red | white）和（red | red），则在图中可以给出回路（green | red）（red | white）（red | red），得到的图不符合题意，不能表示“对象”与“对象”之间的关系。

设图$G(V, E)$，每种颜色对应一个节点，$V=\{ v_1, v_2, \ldots\ldots, v_m \}$，每颗珠子对应一条边，若有珠子$(c1, c2)$，就有无向边$\{v_{c1}, v_{c2}\}$。收集的珠子是否能够联成项链的问题就转化为在图$G$中是否存在欧拉链的问题。

算法步骤如下：

① 在输入测试用例的同时构造图 G；计算每个节点的度数；对于一条边的两个节点，合并节点所在的并查集；找出序号值最小的节点 S；

② 按照递增顺序搜索每个节点所在并查集的根：若存在两个节点分属不同的并查集，则说明图 G 是不连通图，欧拉链不存在；否则，

③ 按照递增顺序分析每个节点的度：若存在度数为奇数的节点，则判定欧拉链不存在，无法做出项链；否则，

④ 从 S 出发，通过 DFS 寻找欧拉链。

4 在《集合与图论》课程中引入联合实验后的教学效果

《集合与图论》联合实验还包括：基于偏序关系的树的遍历；基于贪心算法证明的最大生成树算法实现；基于多叉哈夫曼树正确性证明的算法实现，等等。

《集合与图论》联合实验的实践教学效果，目前从课程建设，以及跨校、跨区域的教学实验体系建设两个方面体现。

《集合与图论》课程，在奠定学生的计算机专业知识体系的理论基础，一方面，磨练学生的思维方式数学化，另一方面，通过联合实验课程，打通计算机理论和程序设计类课程的关联，系统提高学生应用理论解决问题的能力。经过《集合与图论》课程的磨练，同学们在后续“本科生进入实验室”的活动中，思维方式数学化和联合实验的训练，奠定了学术研究的基础，在实验室找到研究切入点后，在本科期间，普遍能在SCI三区、二区发表学术论文。课程《集合与图论》现在也已经是国家级一流本科课程的线下一流课程。

在及跨校、跨区域的教学实验体系建设中，“欧拉图构造性证明的算法实现”和其他的《集合与图论》联合实验作为今年的教育部-华为“智能基座”程序设计课程虚拟教研室和ICPC训练联盟的训练营课程，有来自两岸四地的527所高校，3316名学生参加，反响良好。而《集合与图论》联合实验也可以作为程序设计竞赛选手的解题策略训练课。

实践证明，《集合与图论》联合实验能有效地提高学生综合运用知识编程解决问题的能力。

5 结束语

本文以《集合与图论》与程序设计类课程的联合实验课“欧拉图构造性证明的算法实现”为例，展示《集合与图论》的联合实验课程，以编程解决程序设计竞赛试题的实验为切入点，培养学生综合应用集合与图论、数据结构编程解决问题的能力，体会《集合与图论》与其他课程的关联，并在解题过程中，体会数学定义、定理、证明的精确描述，在实践中进行思维方式数学化的训练。

目前，《集合与图论》的联合实验课程的建设作为教育部产学合作协同育人项目（阿里云支持），在线下授课的同时，也同时在线上向两岸四地授课[14]，并在进一步完善中。

参 考 文 献

[1] 吴永辉, 王建德. 数据结构编程实验：大学程序设计课程与竞赛训练教材（第 3 版）. [M].机械工业出版社. 2021, ISBN 9787111687429.

[2] 周娟, 吴永辉. 程序设计实践入门：大学程序设计课程与竞赛训练教材. [M]. 机械工业出版社. 2021, ISBN 9787111685791.

[3] 吴永辉, 王建德. 算法设计编程实验：大学程序设计课程与竞赛训练教材（第 2 版）. [M].机械工业出版社. 2020. ISBN：978-7-111-64581-8

[4] Wu Yonghui, Wang Jiande. Algorithm Design Practice : for Collegiate Programming Contest and Education. (English Version). [M]. CRC Press. 2018. ISBN 9781498776639

[5] 吳永辉, 王建德. 提升程式設計的資料結構力 第二版：國際程式設計競賽之資料結構原理、題型、解題技巧與重點解析. [M]. 碁峰. 2017. ISBN：9789864765249.

[6] Wu Yonghui, Wang Jiande. Data Structure Practice : for Collegiate Programming Contests and Education. [M]. CRC Press. 2016. ISBN 9781482215397.

[7] 吳永辉, 王建德. 程序设计解题策略：大学程序设计课程与竞赛训练教材. [M]. 机械工业出版社. 2015. ISBN 9787111488316.

[8] 吳永辉, 王建德. 程式設計的解題策略：活用資料結構與演算法. [M]. 碁峰. 2015. ISBN 9789863476818.

[9] Yonghui Wu. The Implementation for Polishing Students' Programming Skills Solving Problems[C]. Proceedings of the 2021 International Conference on Diversified Education and Social Development. Beijing: Atlantis Press. 2021:92-97

[10] Yonghui Wu. Programming Training League: A System Organizing Programming Training Cross Region. [EB/OL]. ICPC Competitive Learning Institute Symposium 2018. [2018-04-18]. https://ciiwiki.ecs.baylor.edu/index.php/ICPC_CLIS_2018,_Beijing,_China

[11] Yonghui Wu. Cooperating Programming Contest Training with Education. ICPC Competitive Learning Institute Symposium 2019. [EB/OL]. [2019-04-02]. https://ciiwiki.ecs.baylor.edu/index.php/ICPC_CLIS_2019,_Porto,_Portugal

[12] 吳永辉. 数据结构编程实验[EB/OL]. 复旦大学在线教育中心(https://mooc1-1.chaoxing.com/course/215508752.html), 2019.

[13] 吳永辉. 程序设计 [EB/OL]. 复旦大学在线教育中心 (https://mooc1-1.chaoxing.com/course/220044602.html), 2021.

[14] 吳永辉. 集合与图论 [EB/OL]. 复旦大学在线教育中心 (https://mooc1-1.chaoxing.com/course/216035861.html), 2022.