

# 《数据结构与算法设计》

## 课程设计总结

学号： 1950084

姓名： 陈泓仰

专业： 计算机科学与技术

2021 年 7 月

# 目 录

第一部分 算法实现设计说明.....	3
1.1 题目.....	3
1.2 软件功能.....	3
1.3 设计思想.....	3
1.4 逻辑结构与物理结构.....	4
1.5 开发平台.....	4
1.6 系统的运行结果分析说明.....	4
1.7 操作说明.....	5
第二部分 综合应用设计说明.....	5
2.1 题目.....	6
2.2 软件功能.....	7
2.3 设计思想.....	7
2.4 逻辑结构与物理结构.....	7
2.5 开发平台.....	7
2.6 系统的运行结果分析说明.....	8
2.7 操作说明.....	8
第三部分 实践总结.....	9
3.1.所做的工作.....	9
3.2.总结与收获.....	9
第四部分 参考文献.....	9

# 第一部分 算法实现设计说明

## 1.1 题目

给定一个图，完成：

- (1) 建立并显示出它的邻接链表；
- (2) 分别用普里姆算法和克鲁斯卡尔算法构造其最小生成树，随时显示其构造的过程；
- (3) 给出某一确定顶点到所有其它顶点的最短路径；
- (4) 给出每一对顶点之间的最短路径。

## 1.2 软件功能

1. 读入一张图和起点
2. 展示邻接表
3. 使用 Prim 和 Kruskal 算法分别计算并生成最小生成树，记录中间结果(加边顺序)，以供展示树的生成过程。
4. 使用 Dijkstra 算法计算单源最短路，记录额外信息以展示最短路径。
5. 使用 Floyd 算法，计算任意两点对间最短距离。
6. 设计简介易用的前端以供输入和展示输出

## 1.3 设计思想

将不同算法封装成独立模块。

采用统一的读入结构读入一张图。再输入想要使用的功能，来调用不同算法模块，输出结果。

本题算法均为“模板题”，已在大二上学期中学习。

为方便后续封装成 dll 模块，通过字符串 (char\*) 的形式将所有信息一次性读入，再转换为 stringstream，依次将信息存入数据结构中。

```
char c[100] = "123 456";
string smp = c;
stringstream ss;
ss << smp;
int a, b;
ss >> a >> b;
cout << a << " " << b << endl; // a = 123(int)  b = 456(int)
```

将所有需要输出的内容向字符串 (string) 保存，最后转换为字符串 (char\*) 向外输出。

```
string fea2_kruskal() {
    string outp = "";
    .....
    outp += to_string(e[i].l) + " " + to_string(e[i].r) + " " + to_string(w) + "\n";
    .....
    return outp;
}
.....
const char* p = outp.c_str();
```

```
cout << p;
```

## 1.4 逻辑结构与物理结构

使用链式前向星（邻接表的一种实现）存储一个图，包括：

表头 `int head[N]`

边 `struct e {int l, r, w, ne;}` // 起点、终点、权值、指向的下一条边的序号

邻接表模块，无额外数据结构。

最小生成树（prim 算法）模块，除了使用上述结构，还包括：

记录每个节点是否被访问到，并判断输入的图是否联通（能构成树）：`bool vis[N]`

对 prim 采用堆优化，提高算法效率：优先队列 `priority_queue<node_prim> q`

\*`node_prim` 记录完整边信息（以供记录并输出加边顺序）；以权值为关键字排序

最小生成树（Kruskal 算法）模块，除了使用上述结构，还包括：

并查集模块，用于快速判断下一条边可否加入生成树中：`int fa[N]`

```
int find(int x) { return fa[x] == x ? x : fa[x] = find(fa[x]); }
```

单源最短路（Dijkstra 算法）模块，除了使用上述结构，还包括：

对算法采用堆优化，提高算法效率：优先队列 `priority_queue<node_prim> q`

记录源点到各个节点的当前最小值：`int dis[N]`

记录每个节点是否被处理过 `bool vis[N]`

\*Dijkstra 算法每个节点入队即为最优解，也由此保证时间复杂度。

多源最短路（Floyd 算法）模块，除了使用上述结构，还包括：

邻接矩阵：`int map[N][N]`（该算法采用邻接矩阵效率更高）

## 1.5 开发平台

本程序采用了跨语言开发，使用的平台包括：

C++：Visual Studio 2019 (Windows, X64 Release)

Python：python3.7.0(测试) 3.8.3(部署) (Windows, Anaconda)

Html/JavaScript/CSS：Edge(Chrome 内核版本) (使用 Html5 ES6 特性)

本程序已部署到我的个人服务器上。访问 <http://ellye.cn:2021/> 即可运行并评测。

其中，在导航栏第一个图标处输入一张图。打开网页即可看到默认数据，可直接点击提交。

随后，点击第 2 到第 5 个图标，即可看到各个算法的实现结果。（最后一个图标对应综合应用相关功能）

# 1.6 系统的运行结果分析说明

考虑到运行效率，我使用 C++完成算法实现，并设计了一个简易的 UI 界面进行测试。

```
int main() {
    init();
    cout << "算法实现\n1 显示该图的邻接链表\n2 分别用普里姆算法和克鲁斯卡尔算法构造其最小生成树，随时显示其构造的过程\n3 给出某一确定顶点到所有其他顶点的最短路径\n4 "
        "给出每一对顶点之间的最短路径\n5\n综合应用\n5 定义一个产品的装配路径，规划拆解和安装路径\n6 请输入要实现的功能：\n";
    cin >> ord;
    .....

    case 5:
        cout << "请输入需要替换的零部件：";
        cin >> s;
        outp = fea5(s);
    }
```

测试完毕后，将其封装成 dll 文件，被 python 接受，进行正确性、稳定性、容错能力测试。

Python 对前端传入的图信息进行处理，包括：

忽略多余空行，和每行多余空格。

对每条边，若第三个参数（权值）未填写或错误（权值非正整数等），给予默认赋值 1。

对每行输入，若参数过少（ $\leq 1$ ）或过多（ $\geq 4$ ），忽略。

每个节点可为字符串名字，不必须为正整数。对每个输入的节点，自动依名字转换为节点编号，再将编号传入（由 C++封装的）dll，给出运行结果后，再将结果中的编号对应回输入的字符串名。

当输入的点数非正整数，或者多余实际识别到的不同点数时，自动修正为实际点数。

当输入的边数非正整数，或者多余实际识别到的边数时，自动修正为实际边数。

在最小生成树中，本程序强制将输入的图识别为无向图。

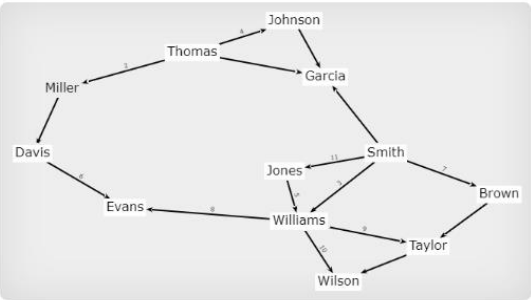
在多源最短路中，采用等宽字体，缩放字号，以列表的形式完整展示所有数据。

# 1.7 操作说明

访问 <http://ellye.cn:2021/>



点数和边数空缺将自动统计；起始点允许为非数字字符。



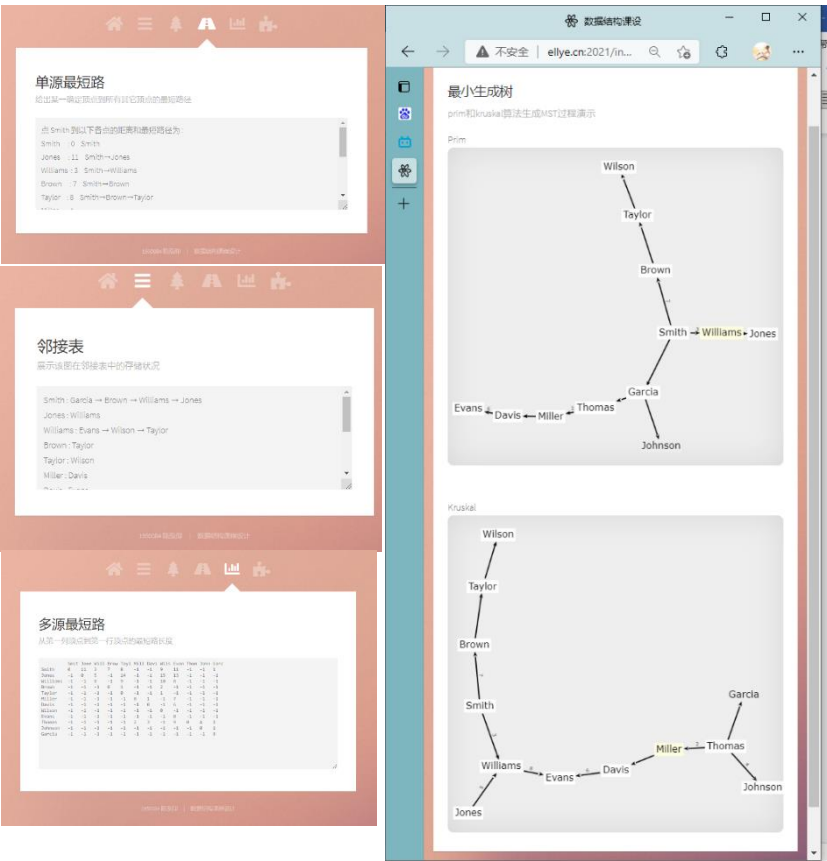
在默认界面中（左上图），输入一张图。其中，点和边数可以省略，由程序自动识别。在第三栏中，若输入无向图，则填写 0。

在较大的框中输入您的图。每一行输入 2 或 3 个参数 A、B、C(=1)，代表一条 A 指向 B，权值为 C 的边。

在最后一栏填写最短路起点，供单源最短路模块使用（可以缺省，默认维修识别到的第一个零件）。

点击提交，即可看到拓扑关系图（右上图）。

点击导航栏第 2-第 5 个图标，即可看到不同功能的实现。



其中，最小生成树采用了动态展示，实时展示两种算法每次加入的边。

## 第二部分 综合应用设计说明

### 2.1 题目

在产品装配过程中，需根据工艺路径，规划产品每一个零部件的先后装配次序。反之，在产品维修时，需根据该装配次序逆向拆解，直至拆下需替换的零部件，重新按先前的拆除次序逆向安装。对于汽车、轨道机车、飞机等复杂产品，由于其零部件众多，维修复杂，现需开发一维修教学系统，模拟其拆解维修过程，请设计其中拆解及安装过程仿真算法，实现产品装配次序输入，通过指定需替换的零部件，规划拆解路径和安装路径。

- (1) 可定义一个产品的装配路径。
- (2) 根据需拆解的零部件，自动规划拆解路径和安装路径。

## 2.2 软件功能

1. 输入一张图
2. 输入需要替换的零件
3. 输出零件拆卸、安装的顺序

## 2.3 设计思想

首先考虑给出一个产品的装配次序。一个节点（零件）允许被装配的前提，是其前置零件已经全部装配完成。对应到拓扑排序，即前置节点全部入队，当前节点的当前入度为 0。不难想到，直接进行拓扑排序可以给出一个合理的顺序。假设有一个产品，拥有 10 个零件，它的一种装配次序为  $1 \rightarrow 2 \rightarrow \dots \rightarrow 10$ 。当需要维修 7 号零件时，只需要将 7 及其后的所有零件拆除 (10 9 8 7)，维修后再依次装回 (7 8 9 10)，即可。

然而，单纯进行一次拓扑排序，给出的结果是可能存在冗余的。考虑第二个产品，拥有 3 个零件。他们的装配关系为： $1 \rightarrow 3$ 、 $2 \rightarrow 3$ 。易得，它的一种装配次序为： $1 \rightarrow 2 \rightarrow 3$ 。按照之前的算法，要维修 1 号零件，我们需要拆开 2 号和 3 号零件；但显而易见，2 号零件是不需要拆开的。

为了解决这个问题，我们先从需要维修的零件节点开始，对整张图做一次遍历（时间复杂度  $O(m)$ ），找出从这个节点可以到达的所有节点，我们称为“相关节点”。最后，再对这些相关节点进行拓扑排序，即可给出最优解，避免拆除额外的、并不需要拆除的零件。

## 2.4 逻辑结构与物理结构

使用链式前向星（邻接表的一种实现）存储一个图，包括：

表头 `int head[N]`

边 `struct e {int l, r, w, ne;} // 起点、终点、权值、指向的下一条边的序号`

找到和这个零件相关的所有零件，除了使用上述结构，还包括：

访问队列 `queue<int> q`

记录每个节点是否被访问的数组 `vis[N]`

对所有相关零件进行拓扑排序，除了使用上述结构，还包括：

记录每个顶点入度 `deg[N]`

## 2.5 开发平台

本程序采用了跨语言开发，使用的平台包括：

C++：Visual Studio 2019 (Windows, X64 Release)

Python：python3.7.0(测试) 3.8.3(部署) (Windows, Anaconda)

Html/JavaScript/CSS: Edge(Chrome 内核版本) (使用 Html5 ES6 特性)

本程序已部署到我的个人服务器上。访问 <http://ellye.cn:2021/>即可运行并评测。

其中，在导航栏第一个图标处输入一张图。打开网页即可看到默认数据，可直接点击提交。

随后，点击最后一个图标，即可看到运行结果。（其他图标对应算法实现相关功能）

## 2.6 系统的运行结果分析说明

因为前端部分和算法实现共用，仅说明和 1.6 节中不同处：

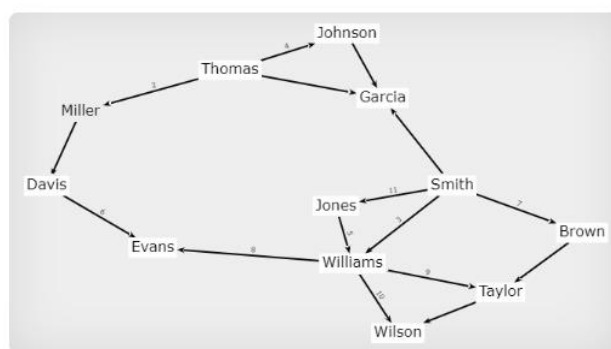
在综合应用中，本程序忽略用户指定，强制将输入的图识别为有向图。当输入依然不符合拓扑排序要求（成环）时，输出其中一个环，提醒用户检查数据。

## 2.7 操作说明

访问 <http://ellye.cn:2021/>



点数和边数空缺将自动统计；起终点允许为非数字字符。



在默认界面中（左上图），输入一张图。其中，点和边数可以省略，由程序自动识别。本综合应用强制将输入识别为有向图，不需要填写是否有向。

在较大的框中填写零件组装顺序。每一行输入两个参数 A、B，代表安装零件 B 前必须先安装零件 A。（可以直接运行默认数据）。

在最后一栏填写想要维修的零件（可以缺省，默认维修识别到的第一个零件）。

点击提交，即可看到拓扑关系图（右上图）。

点击导航栏最后一个图标（拼图形状），即可看到拆卸和组装顺序。





## 第三部分 实践总结

### 3.1.所做的工作

算法：C++完成算法模块。

后端：利用 python，将前端传入的数据高效处理，保证程序健壮性；随后将处理好的数据传入算法模块，将返回的结果加上人性化提示后返回至前端。

前端：抱着让程序可以在多设备上运行的想法，本程序以网页(HTML)形式呈现，各客户端（电脑/手机）均可访问本软件。同时，本程序采用响应式设计（使用模板）来保证各端显示界面优美。JS 方面，使用 JQuery 和 Springy.js 以提高编写效率。

在软件程序全部编写完成后，我学习了服务器相关知识，将项目布置到了服务器，以便于老师和助教评阅时使用，无需录制视频、安装环境或打包 exe。

\*程序网址 <http://ellye.cn:2021/>，可直接访问。

### 3.2.总结与收获

在大学前两年的学习，主要以打基础为主；而本次课程设计的重点，便是将我们学习的算法应用在实际项目中。

在编写实用软件的过程中，产生的新的挑战有：

交互（输入输出）的方式变化：通过网页表单传入，或上传文件到后端读入等。

界面设计：需要对用户友好。

程序健壮性：对各种错误输入的判断和拦截。

我初次体验了 C++-python-HTML/JS 的跨语言编程。

各个语言其特性导致适合完成不同的工作。例如，C++能高效的执行一个算法，而 python 用于数据处理更为方便与高效。因此，一个实用软件通常会涉及不同语言的编写、交互与测试。在本次认知实习中，我先使用 C++完成了算法实现，测试完毕后封装成 dll，通过字符串形式与 python（通过 ctypes 模块）交互，再由 python 处理成 json 与前端（js）交互并显示。

最后，通过本次认知实习，我查找资料的能力也得到了锻炼。对于计算机学科，广袤的互联网总是最好的，终生的老师。

## 第四部分 参考文献

*ctypes* --- Python 的外部函数库 — Python 3.9.7 文档

<https://docs.python.org/zh-cn/3.9/library/ctypes.html>

*ctypes* 官方文档

*python 调用 c++ dll 动态库 - 吓死你个管理员 - 博客园 (cnblogs.com)*

<https://www.cnblogs.com/xiao-yuyy/p/11435480.html>

*Visual Studio 下 c++生成dll 并与python 交互*

*解决 Python 调试 OSError: [WinError 193] %1 不是有效的 Win32 应用程序\_vampl 的博*

客-CSDN 博客

<https://blog.csdn.net/vample/article/details/88877745>

跨语言编程的 Debug

*Springy - A force directed graph layout algorithm in JavaScript. (getspringy.com)*

<http://getspringy.com/>

通过前端显示树的 JavaScript 开源算法

*HTML5 - 让 Canvas 大小自适应（自动撑满整个屏幕） (hangge.com)*

[https://www.hangge.com/blog/cache/detail\\_1871.html](https://www.hangge.com/blog/cache/detail_1871.html)

jQuery 监听代码

*flask 给模板的 js 传参\_FanMagicOne 的博客-CSDN 博客*

[https://blog.csdn.net/qq\\_42091045/article/details/104204766](https://blog.csdn.net/qq_42091045/article/details/104204766)

*HTML5 UP! Responsive HTML5 and CSS3 Site Templates*

<https://html5up.net/>

使用的网页模板

*Springy - A force directed graph layout algorithm in JavaScript. (getspringy.com)*

<http://getspringy.com/>

将输入的图可视化的开源项目 springy.js