

《数据结构》课程设计总结

学号： 1651047

姓名： 杨文翔

专业： 计算机科学与技术

目录

目录..... 2

第一部分 算法实现设计说明..... 4

1. 题目..... 4

 二叉树的建立和删除..... 4

2. 软件功能..... 4

3. 设计思想..... 4

4. 逻辑结构与物理结构..... 5

5. 开发平台..... 5

 开发平台..... 5

 开源代码..... 5

 运行环境..... 6

6. 系统的运行结果分析说明..... 6

 开发及调试过程..... 6

 达到的效果..... 6

 具体案例..... 6

7. 操作说明..... 8

第二部分 综合应用设计说明..... 9

1. 题目..... 9

2. 软件功能..... 9

3. 设计思想..... 10

4. 逻辑结构与物理结构..... 11

5. 开发平台..... 11

 开发平台..... 11

 开源代码..... 12

 运行环境..... 12

6. 系统的运行结果分析说明.....	12
开发及调试过程.....	12
达到的效果.....	13
具体案例.....	14
7. 操作说明.....	16
第三部分 实践总结.....	17
1. 所做的工作.....	17
总结与收获.....	17
第四部分 参考文献.....	18

第一部分 算法实现设计说明

1. 题目

二叉树的建立和删除
输入一组关键值，建立相应的二叉排序树，完成节点的查找和删除操作。
要求：

- 1) 可以实现删除根节点、叶子节点以及其他任意节点的功能；
- 2) 可随时显示操作的结果

2. 软件功能

可以实现删除根节点、叶子节点以及其他任意节点的功能；可随时显示操作的结果。
程序界面：

程序的交互有三种方式：在左侧两行表单的输入框中填入欲操作的数据，然后点击右侧按钮，分别可实现二叉树的数值插入和删除。第三种交互是点击二叉树的结点，可将这个节点下的全部子节点折叠显示。

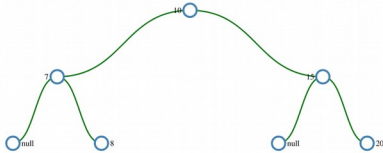


Figure 1: 初始界面

3. 设计思想

核心数据结构由二叉树和节点组成。

二叉树：

```
class BinarySearchTree {
  constructor (node) {
    this.root = node
  }
  insert (node, root) {}
  remove (key, root) {}
}
```

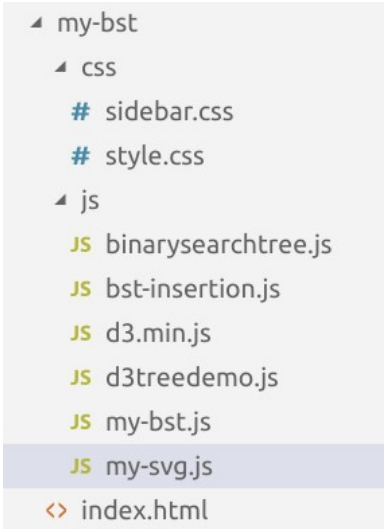


Figure 2: 工作目录

节点：

```
class Node {  
  constructor (value) {  
    this.value = value  
    this.parent = null  
    this.left = null  
    this.right = null  
  }  
}
```

实现的二叉树算法有结点的插入、查找、删除。

插入操作的参数是插入节点和插入的数值。分为三种情况。一，数值等于当前节点，因节点已存在而插入失败，直接返回；二，数值小于当前节点，此时若当前节点无左子树，插入节点为左子树，更新父节点的引用，否则将数值递归地调用左子树的插入操作；三，数值大于当前节点，和第二种情况类似，但用右子树代替操作中的左子树。

查找操作已在插入操作中实现，若插入操作返回插入失败，则查找成功，反之亦然。

删除操作，首先执行搜索，找到欲删除节点，然后分三种情况讨论。一，欲删除节点没有子节点，直接删除此节点，并解除父节点的引用；二，欲删除节点有一个子节点，只需用这个子节点替代当前节点，并更新父节点的引用；三，欲删除节点有两个子节点，这时需要用当前节点子树中次大的元素替代当前节点。具体操作是先取得右子节点，然后一直递归地获取左子树，走到叶节点，然后用这个叶节点替代当前节点，并删除这个叶节点，最后更新父节点的引用。

4. 逻辑结构与物理结构

数据结构采用的逻辑结构是二叉树，每个结点记录对父节点和双子节点的引用。物理结构为链式存储。二叉树除了链式储存，还可使用数组连续储存，充分发挥内存存在连续空间中的读写性能。此项目中选择使用链式储存，原因是链式储存可以避免不平衡二叉树带来的空间浪费。这在运行于占用内存量大的 Chrome 内核时更能发挥优势。

5. 开发平台

开发平台

- Linux 操作系统，Archlinux 发行版（2018.9.4 最新滚动更新）
- [Visual Studio Code](#)

开源代码

- 数据可视化：[D3](#) Javascript 库

- 前端设计: [jQuery](#) 及 [Bootstrap](#)
- 字体: [popper](#)

运行环境

因为程序运行时使用 Electron 包装，运行要求等同于 Electron 运行要求。虽然 Electron 未提供 RAM 与 CPU 要求，由于 Electron 使用 Chromium 内核，故硬件要求等同于 Chromium 运行要求。

- Windows 7 及以上
- 提供 X86 及 X64 二进制文件，不支持 ARM 架构
- 具有 SSE2 指令集的 CPU，其中英特尔 CPU 要求奔腾 4 及以上
- 512MB RAM
- 连接互联网（为了加载字体和 CSS）

6. 系统的运行结果分析说明

开发及调试过程

主要开发工具有：文本编辑器：VSCode；调试工具：Chrome 开发者工具。

开发过程为，创建本地目录结构，分为 html、js、css 目录，使用文本编辑器编写代码，在浏览器中可事实看见编辑效果
Chrome 开发者工具提供 Javascript 的 REPL 功能，随时查看变量内容，方便调试。

达到的效果

软件左侧控制区域具有 Bootstrap 美化效果，右侧绘图区域可实时相应用户交互，如在点击节点时收起、展开二叉树；所有操作，包括插入、删除、发生错误、收起展开节点时都有渐变动画。

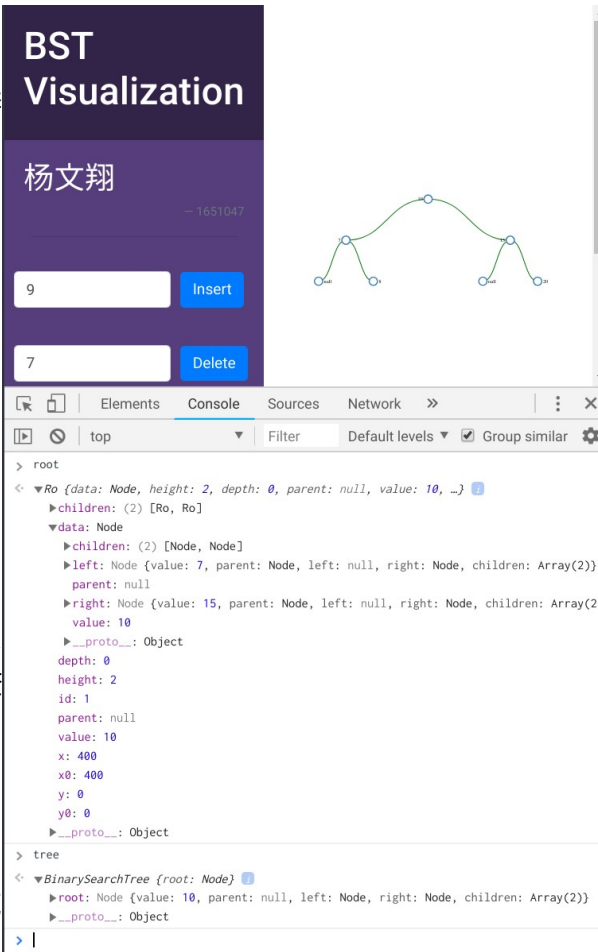
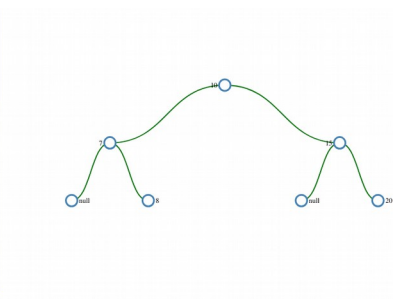
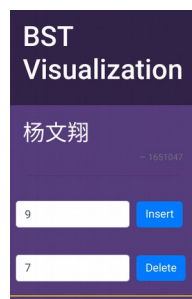


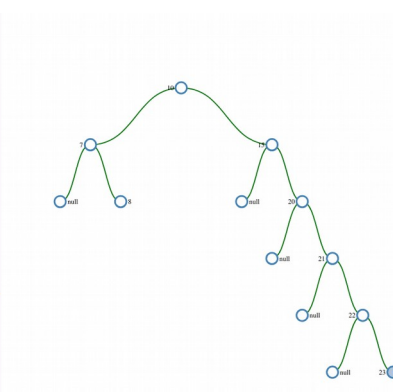
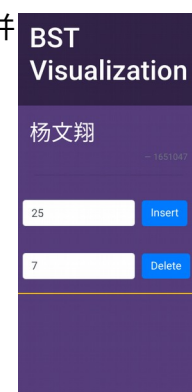
Figure 3: 实时调试

具体案例

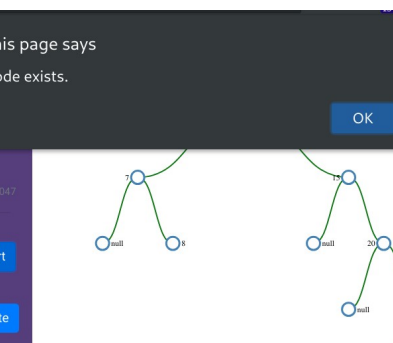
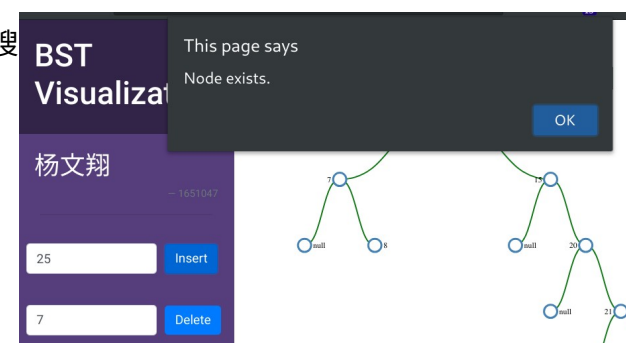
1. 启动软件，软件展示预先定义的二叉树。



2. 连续插入数值，正确以二叉搜索树的插入顺序插入并显示。



3. 插入已存在数值，弹出错误提示，表示在二叉树中搜索到插入数值。



7. 操作说明

运行步骤：

1. 使用浏览器（Chromium 68.0.3440.106 已测试）打开 index.html 文件



Figure 3: 初始状态

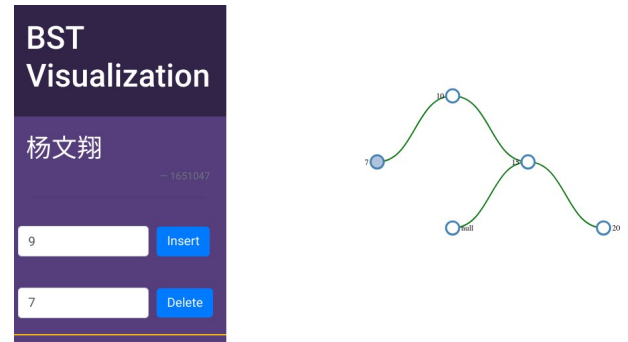


Figure 4: 任意点击节点，节点收起，隐藏整个子树。

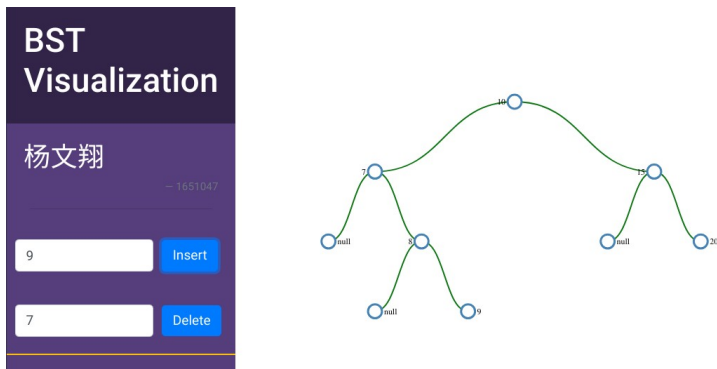


Figure 5: 在插入表单中填入 9，点击插入（Insert），左侧节点被展开，值为 9 的节点插入。

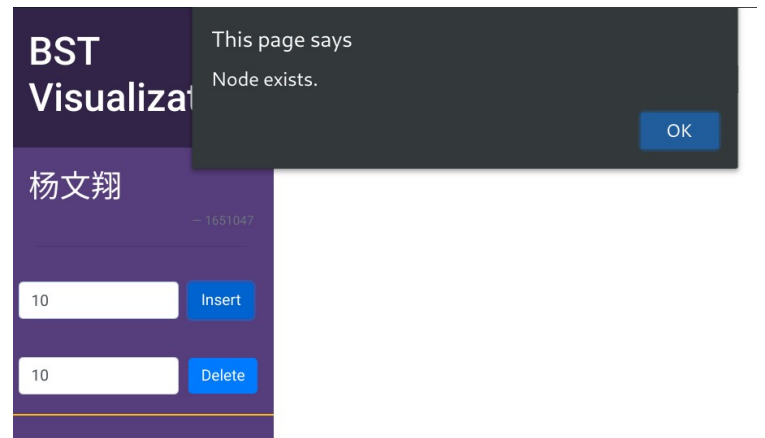


Figure 4: 使用删除表单删除根结点，根结点被删除，然后使用插入表单中再次插入已存在的节点，弹出错误提示

第二部分 综合应用设计说明

1. 题目

参加奥运会有 n 个国家，各国编号为 $1.....n$ ，比赛分成 m 个男子项目，和 w 个女子项目。项目编号为男子 $1.....m$ ，女子 $m+1.....m+2$ ，不同的项目取前五名或前三名积分；取前五名的积分分别为：7、5、3、2、1，前三名的积分分别为：5、3、2；哪些取前五名或前三名由学生自己设定。

- (1) 可以输入各个项目的前三名或前五名的成绩；
- (2) 能统计各国总分；
- (3) 可以按各国编号、各国总分、男女团体总分排序输出；
- (4) 可以按各国编号查询国家某个项目的情况，可以按项目编号查询取得前三或前五名的国家。

2. 软件功能

- (1) 可以输入各个项目的前三名或前五名的成绩；

程序初始化后，自动填充一系列初始化数据，方便展示。在程序顶部的 Create 标签下，有以国家编号和名称插入、删除国家，以及把向国家颁发某项比赛的奖的功能。

实现方式：每次增加国家，向表示国家列表的数据结构插入使用用户输入的名称初始化的国家实例，删除国家时亦然。将国家与比赛绑定时，首先分别在选中国家和比赛的实例中记录这次绑定，然后更新国家实例中表示总分的变量。

- (2) 能统计各国总分；

在程序顶部的 Summary 标签下，展示国家列表和国家获得的分数，包括男女团体得分。

实现方式：对表示国家的数据结构进行遍历，显示国家名。对每个国家，对其获得奖项的比赛进行遍历，求和统计其分数和男女团体总分，显示国家总分。

- (3) 可以按各国编号、各国总分、男女团体总分排序输出；

在程序顶部的 Ranking 标签下，以 Radio Button 的形式为用户提供排序方法的选择。在用户作出选择后，右侧的表格作出相应的回应，以用户选择的排序方法展示相应信息。可选的排序方法有：国家 ID、国家名字、国家分数、男女团体总分。

实现方式：后端通过 HTTP GET 请求的参数获取用户对排序方式的不同选择。对内存中的国家实例列表以用户选择的有序参数为排序依据，排列后返回表格。

(4) 可以按各国编号查询国家某个项目的情况，可以按项目编号查询取得前三或前五名的国家。

在程序顶部的 Querying 标签下，以 Radio Button 的形式为用户提供查询国家还是项目的选择，然后用户在输入框中输入所选国家或项目的编号，点击查询按钮，右侧表格依照用户选择列出相关信息。

实现方式：后端通过 HTTP GET 请求的参数获取用户对查询条件的不同选择，然后通过返回对应的表格将结果呈现给用户。

3. 设计思想

实现思路：因为以表单形式接受用户输入、反馈表格输出，使用 Express (NodeJS) 进行后端编成，使用 Bootstrap 设计前端界面。

数据结构选择：题目主要是国家和比赛项目的信息关联问题，类似数据库的查询操作，故选择分别为国家和比赛创建类，并依靠每个实例自行记录自己相关信息的方式维持程序状态的连续。

```
class Sport {
  constructor(id, name, numAwards, gender) {
    this.id = id
    this.name = name
    this.numAwards = numAwards
    this.countries = []
    this.gender = gender
  }

  insert(country) {
    if (this.countries.length < this.numAwards) {
      this.countries.push(country)
      country.award(this)
    }
  }

  clear() {
    this.countries = []
  }
}
```

```

class Country {
  constructor(id, name) {
    this.id = id
    this.name = name
    this.awards = []
    this.manScore = 0
    this.womanScore = 0
    this.score = 0
  }

  award(sport) {
    this.awards.push(sport)
    this.updateStats()
  }

  updateStats() {
    this.manScore = scoreCountry(this, 'man')
    this.womanScore = scoreCountry(this, 'woman')
    this.score = this.manScore + this.womanScore
  }
}

```

比赛类和国家类都需要通过类方法实现插入和更新状态。插入过程如下：对于预先定义好的比赛列表，国家获得其中一个比赛的奖，在调用 `sport.insert(country)` 时，首先将国家记录在这项比赛的记录中，这个方法也保证获奖国家数量不超过预先定义好的 7 个或 5 个；然后 `insert` 方法会再反过来调用国家的 `award` 方法，通知国家实例记录自己获得的奖项，并且更新自己获得的分数。

4. 逻辑结构与物理结构

国家和比赛的实例都以数组的逻辑结构保存引用，因为数组具有高速随机读取的优点，有利于提升程序性能，而国家和比赛的数目不经常修改，因此以储存引用的数组无需频繁修改，避开数组难以删除和插入的缺点。国家和比赛的类都必须调用类方法对成员变量进行更新，这提升了代码的封装程度，降低出错的可能。

5. 开发平台

开发平台

- Linux 操作系统，Archlinux 发行版（2018.9.4 最新滚动更新）
- [Visual Studio Code](#)

开源代码

- 后端: [Nodejs 10.10.0](#) 及 [Express 4.16.3](#)
- View Engine: [PUG](#)
- 前端设计: [jQuery](#) 及 [Bootstrap](#)

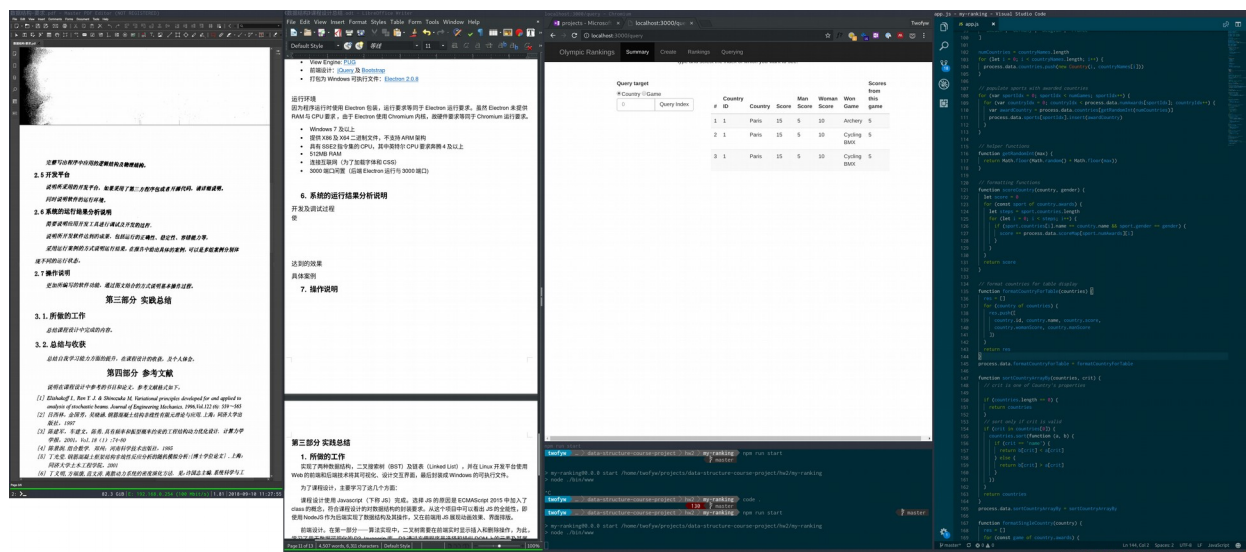
运行环境

因为程序运行时使用 Electron 包装, 运行要求等同于 Electron 运行要求。虽然 Electron 未提供 RAM 与 CPU 要求, 由于 Electron 使用 Chromium 内核, 故硬件要求等同于 Chromium 运行要求。

- Windows 7 及以上
- 提供 X86 及 X64 二进制文件, 不支持 ARM 架构
- 具有 SSE2 指令集的 CPU, 其中英特尔 CPU 要求奔腾 4 及以上
- 512MB RAM
- 连接互联网 (为了加载字体和 CSS)
- 3000 端口闲置 (后端 Electron 运行与 3000 端口)

6. 系统的运行结果分析说明

开发及调试过程



NodeJS 程序开发, 使用 nodemon 对开发目录进行文件监控, 在工作目录发生任何文件修改时, 自动重新启动 node 网页服务器。用 VSCode 的 Debug 功能, 挂载 node 进程, 可以下断点、监控变量数据。

达到的效果
程序效果与预期相符。

具体案例

Olympic Rankings					
Summary					
Create					
Rankings					
Querying					
#	Id	Country	Score	Woman Score	Man Score
1	0	Athens	2	0	2
2	1	Paris	18	2	16
3	2	United States	8	8	0
4	3	United Kingdom	1	1	0
5	4	Sweden	36	36	0
6	5	Germany	0	0	0
7	6	Belgium	7	0	7
8	7	France	6	5	1

Figure 6: 程序初始化后，自动填入初始化数据，并在 summary 页面列出

Olympic Rankings					
Summary					
Create					
Rankings					
Querying					
We have 8 countries and 3 games, select the query type and select the index of which you want to see.					
Query target					
Country Game					
0 Query Index					
#	Game ID	Game	Number of Prices	Awarded to	Game Type
1	1	Boxing	5	Sweden	woman
2	1	Boxing	5	United States	woman
3	1	Boxing	5	Sweden	woman
4	1	Boxing	5	Sweden	woman
5	1	Boxing	5	United Kingdom	woman

Figure 7: 在 Querying 页面，按用户选择的查询类型和编号展示查询结果

Olympic Rankings

Summary

Create

Rankings

Querying

Country Name

China

Add Country

Select a Sport

0

Awarded country ID

0

Add award

#	Sport	Number of Awarded Countries
0	Archery	5
1	Boxing	5
2	Cycling BMX	3

Clear Awards

Figure 8: 在 Create 页面，可以增添国家、清空国家列表、建立新的国家和比赛的连接，并实时更新表格

运行步骤：

- The figure illustrates the initial state of a web application for Olympic Rankings. It consists of two main parts: a data table on the left and a search interface on the right.

Left Panel: Olympic Rankings Table

#	id	Country	Score	Woman Score	Man Score
1	0	Athens	2	0	2
2	1	Paris	18	2	16
3	2	United States	8	8	0
4	3	United Kingdom	1	1	0
5	4	Sweden	36	36	0
6	5	Germany	0	0	0
7	6	Belgium	7	0	7
8	7	France	6	5	1

Right Panel: Search Interface

At the top, there are navigation tabs: Olympic Rankings, Summary, Create, Rankings, and Querying. Below these, a text box states: "We have 8 countries and 3 games, select the query type and select the index of which you want to see."

The search interface includes a "Query target" section with two radio buttons: "Country" (selected) and "Game". Below these is a text input field containing "0" and a "Query Index" button.

Below the search interface is a table showing the results of the search:

#	Game ID	Game	Number of Prices	Awarded to	Game Type
1	1	Boxing	5	Sweden	woman
2	1	Boxing	5	United States	woman
3	1	Boxing	5	Sweden	woman
4	1	Boxing	5	Sweden	woman
5	1	Boxing	5	United Kingdom	woman

Figure 10: 在 Querying 页面，按用户选择的查询类型和编号展示查询结果

Figure 11: 在 Create 页面，可以增添国家、清空国家列表、建立新的国家和比赛的连接，并实时更新表格

第三部分 实践总结

1. 所做的工作

实现了两种数据结构，二叉搜索树（BST）及链表（Linked List），并在 Linux 开发平台使用 Web 的前端和后端技术将其可视化、设计交互界面，最后封装成 Windows 的可执行文件。

为了课程设计，主要学习了这几个方面：

课程设计使用 Javascript（下称 JS）完成。选择 JS 的原因是 ECMAScript 2015 中加入了 class 的概念，符合课程设计的对数据结构的封装要求。从这个项目中可以看出 JS 的全能性，即使用 NodeJS 作为后端实现了数据结构及其操作，又在前端用 JS 展现动画效果、界面排版。

前端设计。在第一部分——算法实现中，二叉树需要在前端实时显示插入和删除操作，为此，学习了用于数据可视化的 D3 Javascript 库。D3 通过方便程序员选择和操纵 DOM 上的元素及其属性，和数据与前端元素的绑定和更新，使得利用前端技术实现数据可视化更为简单。第二部分——综合应用设计中，网页前端使用 PUG view engine 替代 HTML，令代码开发和阅读更清晰简单。利用 Bootstrap 的网格系统进行动态页面布局。

后端设计。在第二部分——综合应用设计中，为了在多页面中保存状态，使用后端技术为多个 HTML 页面提供 RESTFUL API，学习了 NodeJS 和 Express MVC 框架。

总结与收获

虽然课程设计只有看似算法设计和综合应用实现两个题目，但是从布置课程设计题目到完成的整个过程中，不仅仅是学习到了以前未曾接触过的标准用户交互程序，从简陋的控制台程序提升到界面友好、美观、人性化的应用程序，打开了编程的一扇大门，同时也从心态、自学能力、解决未知问题能力等内在的综合实力有了全新的提升。具体来说，学习了使用 Javascript 做前后端变成实现网站的编程框架，包括基本控件的掌握、绘图框架、模型编程、前后端之间的数据传递等相关知识；学习到了 Github 开源项目托管平台的使用，将其运用到了实际的项目。在软实力上，打开了网页编程的大门，激发自身对编程的热情和后面学习的动力；通过英文自学实现了课程设计，自身的学习能力得到了有效锻炼；开发的过程中，无论是设计还是编程，都会遇到各种各样的问题，通过独立解决这些疑难杂症，在独立解决问题的能力方面也得到了很大提升；最重要的是，树立不惧所有未知知识、解决疑难问题、接受各种挑战的信心。

个人体会。在学习和实现的过程中体会和领悟还是比较的多，这里挑出感触最深的几点吧。第一，做事情一定要找最重要的事情先做，这样做事情才有效率，不拖拉。由于个人比较追求完美，对程序的用户界面和一些细节很是用心，但是这样就好浪费很多时间，效率不高，应该先做好最重要的事情，例如最不可缺少的功能实现，主体实现然后细节修改，而不是本末倒置。第二，学到的东西不一定都用的到，所以开发还是有一些功利心较好，用到什么然后去学习相应的知识并用

到实处，这样才能学以致用，恰到好处。在一开始的学习中，每天看书看教程，发现过去了几天还是什么都不会，然后边学边做，才领悟到学习方法和要点。

第四部分 参考文献

- <http://research.cs.queensu.ca/~jstewart/applets/bst/bst-property.html>
- https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs