

数据结构课程设计报告

田文睿 | 2018/8/30

第一部分 算法实现设计说明

1.1 题目

8.堆的建立和筛选

输入一组关键值，建立相应的二叉排序树，完成结点的查找和删除操作。

要求：

- (1) 可以实现从小到大排序，输出并显示结果；
- (2) 随时显示输出堆顶元素，进行重新筛选后的堆。

1.2 软件功能

软件的功能应该包括：

- (1) 堆的输入；
- (2) 堆的计算；
- (3) 堆的显示：

1.3 设计思想

使用 js 语言设计，算法思路与 c++基本相同，前端使用 html5 canvas 画布输出堆的图形化演示界面。注意 js 中没有 sleep 特性，要实现算法演示过程中的演示需要用的 ES5 的 await 和 async 函数。

算法实现的基本流程为：先将用户输入的堆以“,”号分割，转化为数字存到数组中。在从数组的最后一个变量的父节点开始算出最大堆，直到整个数组都转化为最大堆。将[o]元素也就是现在的最大值放到数组末尾，再对剩下 i-1 个元素求最大堆，如此循环，直到只剩下一个元素。

1.4 逻辑结构及物理结构

浏览器

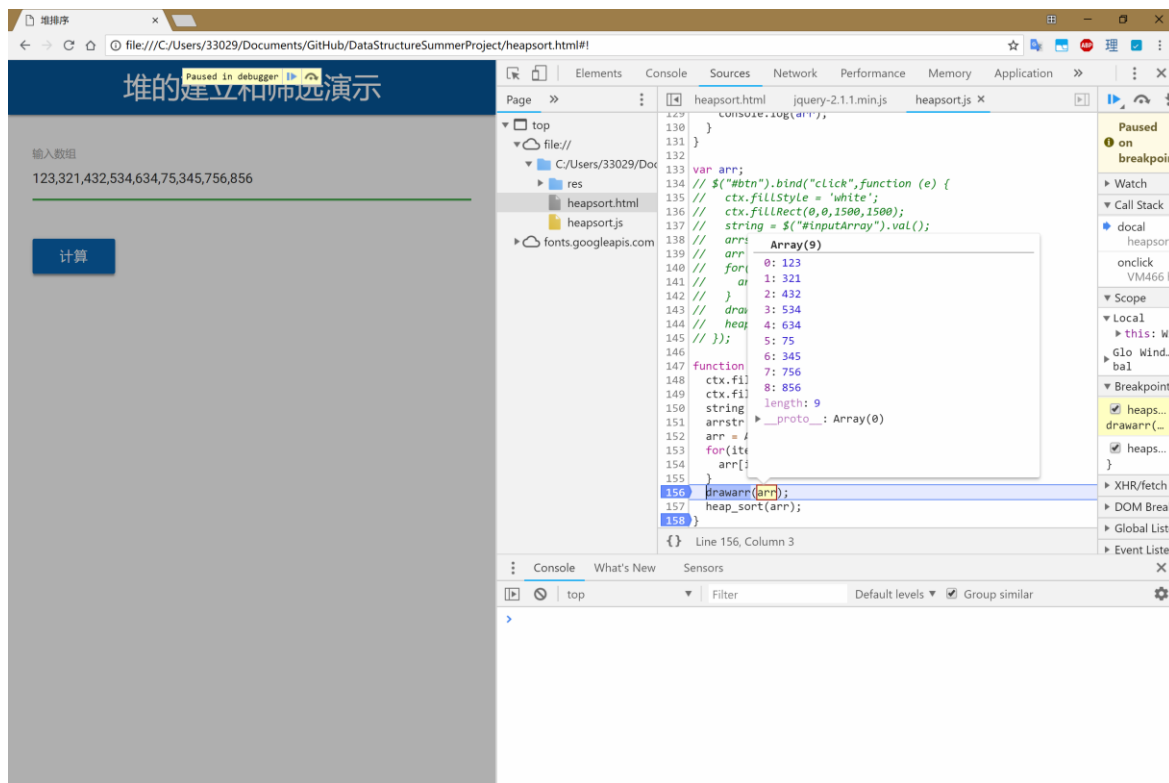
```
|---用户输入堆，点击计算
|      |---btn.onclick()
|      |      |
|      |      |准备 canvas 画布 ctx
|      |      |将用户输入的字符串转换为数组 arr
|      |      |--- drawarr(arr)
|      |      |      |画出起始状态的数组
|      |      |---heap_sort(arr)
|      |      |      |i 等于最小父节点，循环计算最大堆<-----|
|      |      |      |---swap(arr,o,i)
|      |      |      |      |调换 o,i 元素
|      |      |      |      |---drawswap(arr,o,i)
|      |      |      |      |      |画出调换过程
|      |      |      |---max_heapfiy(arr,start,end)
|      |      |      |      |计算出子序列的最大堆
|      |      |      |      |---swap(arr,a,b)
|      |      |      |      |      |调换 a,b 元素
|      |      |      |      |      |---drawswap(arr,a,b)
|      |      |      |      |      |      |画出调换过程
|      |      |      |i<o 循环结束计算完成>-----|
```

1.5 开发平台

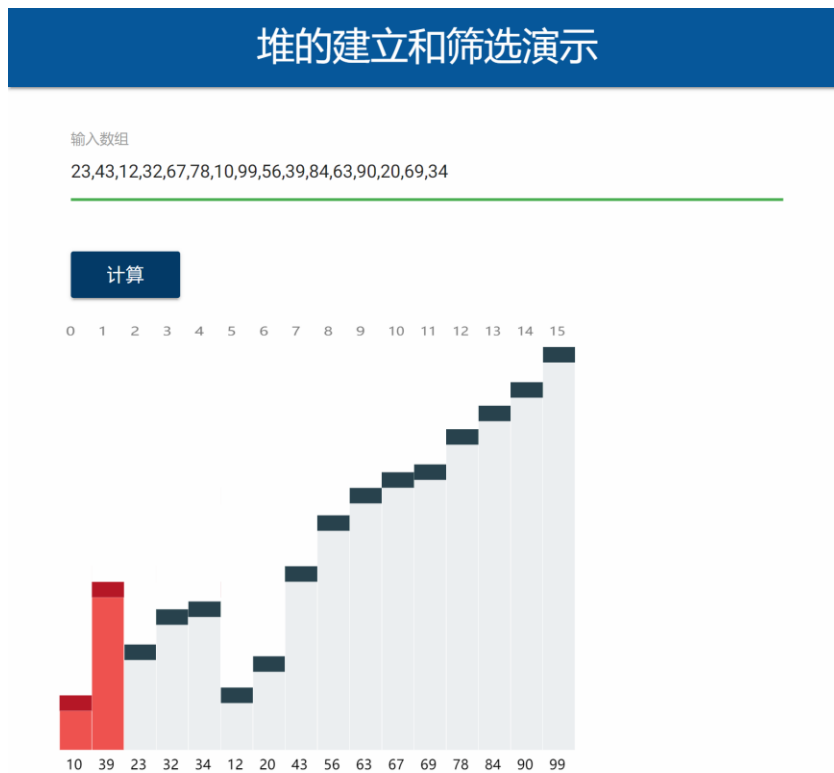
- HTML5
- Javascript ES6
- Jquery (js 扩展库 <https://jquery.com/>)
- Materializecss(开源 css 样式集 <http://www.materializecss.cn/>)
- Exe 文件使用 nodejs 与 electron 编译 (<https://www.electron.build/>)
- 运行环境：支持 html5 canvas 的浏览器或任意 windows10 电脑

1.6 系统的运行结果分析说明

使用 chrome 开发者工具对代码进行调试



代码工作正常

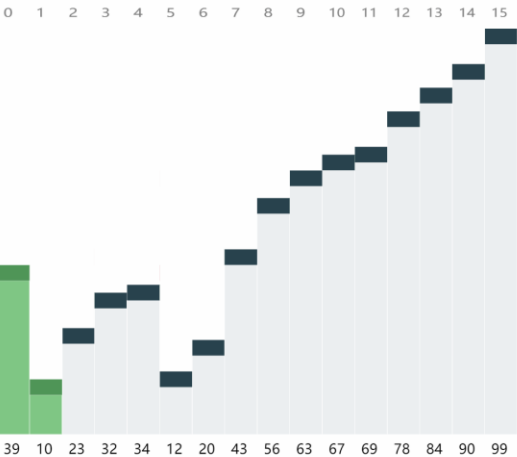


堆的建立和筛选演示

输入数组

23,43,12,32,67,78,10,99,56,39,84,63,90,20,69,34

计算



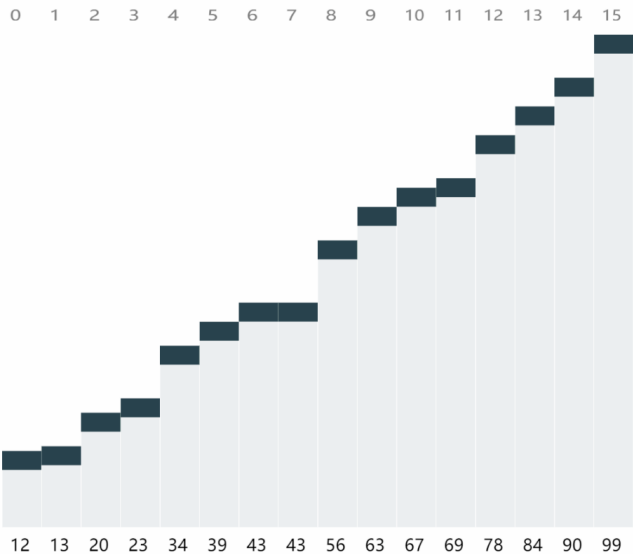
运行结果

堆的建立和筛选演示

输入数组

23,43,12,43,67,78,13,99,56,39,84,63,90,20,69,34

计算



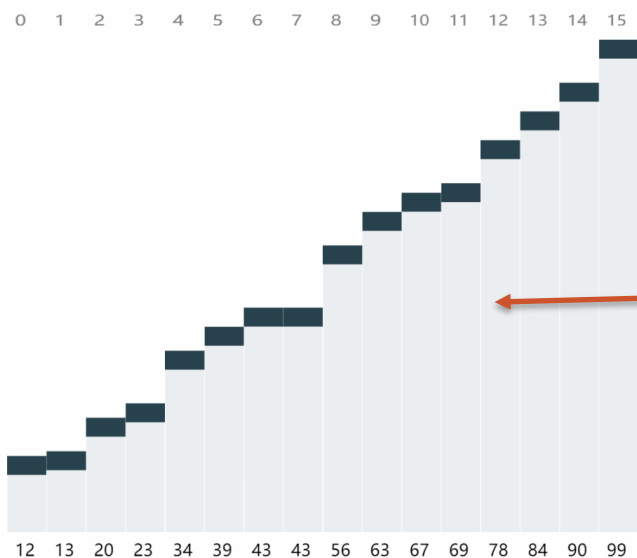
1.7 操作说明

堆的建立和筛选演示

输入数组

23,43,12,43,67,78,13,99,56,39,84,63,90,20,69,34

计算



在此输入堆数组，以，间隔

点此开始计算

计算结果会实时显示在此

第二部分 综合应用部分设计说明

2.1 题目

现需开发一个简单的文本编辑器（支持 200 字以内的输入即可），能支持输入数据的形式包括和范围包括大写、小写的英文字母、任何数字和标点符号。改编辑器需要实现以下功能：

- (1) 对输入的文字进行统计，统计出文字、数字、空格的个数；
- (2) 统计某一字符串在文章中出现的次数，并输出该次数；
- (3) 删除某一子串，并将后面的字符前移；
- (4) 保存文本修改内容，可以撤销修改和恢复修改。

2.2 软件功能

软件应该具有：

- 可以输入文字的输入框
- 显示文字、数字、空格统计的数据
- 保存修改数据，撤销和恢复的按钮
- 统计某一字符串的功能，给用户输入要搜索的字符串，自动显示出现次数

2.3 设计思想

使用 js+html 实现，最后用 nodejs 的 electron 编译成 windows 下的可执行程序。文字输入框用 textaera 实现，用 jquery 给 textarea 绑定 input 触发匿名函数，当用户在 textarea 中输入文字时，匿名函数被触发记录下用户输入的文字的变化，存在一个全局的数组中，因为 js 的数组长度是不固定的，每次只需要 `undo.push(text)` 就可以存入新的文字。当用户点击撤销时就从 `undo.pop()` 获得之前修改的记录，同时将撤销的记录 `push()` 一个 redo 数组中，以供用户恢复撤销的文字。同时在用户每次修改时设置计时器，计时器超时，即

用户在一段时间后没有修改文字，程序自动将文字保存至服务器，使用一个 ajaxpost 将 text 发送到服务器的上的 php 之后 php 再将文字保存在本地一个记录文件上，这样可以做到用户实时修改实时保存。此外程序还有单独保存文件的功能，也是用 ajax post 将文字发送到服务器上的 php 脚本，由 php 脚本保存到数据库中。同时程序启动时会自动发 ajax get 来从服务器获取这些数据。

2.4 逻辑结构和物理结构

浏览器

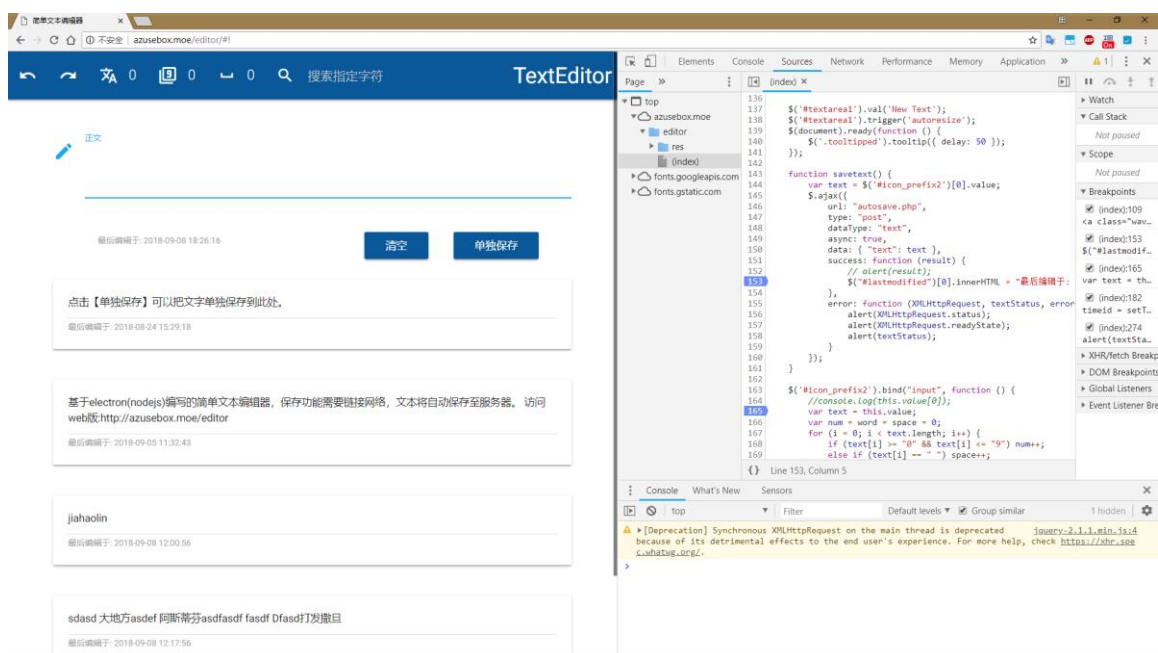
```
|---$.ajax()
|       |获取先前保存在文件中的文字 (正在编辑)
|---$.ajax()
|       |获取先前保存在数据库中的文字 (已保存)
|
|---用户输入文字,
|       |
|       |---input.function()
|       |       |
|       |       |记录撤回数据
|       |       |开始计时器
|       |       |--- savetext()
|       |       |   |计时器超时时保存编辑
|       |---search()
|       |   |搜索指定文字
|       |---undo()
|       |   |撤回, 添加恢复数据
|       |---redo()
|       |   |恢复撤回的数据
|       |---save()
|       |   |将文字单独保存至数据库
```


2.5 开发平台

- HTML5
- Javascript ES6
- Jquery (js 扩展库 <https://jquery.com/>)
- Materializecss(开源 css 样式集 <http://www.materializecss.cn/>)
- Exe 文件使用 nodejs 与 electron 编译 (<https://www.electron.build/>)
- 运行环境：支持 html5 canvas 的浏览器或任意 windows10 电脑

2.6 系统的运行结果分析说明

调试过程：使用 chrome 开发者工具调试，查看 js 运行中的问题和后端 php 的执行情况是否正确。



软件达到的成果：正确处理用户输入的文字，可以稳定将文字保存到服务器。



软件运行结果:

2.7 操作说明

撤销 重做 文字个数 数字个数 空格个数 搜索子串



在此输入文字

正文
这是一个简单的文本编辑器
逻辑基于js编写 前端使用html5 可执行文件使用electron编译
只要在此输入文字 程序便会自动保存至云端
可在必要之时当作一个云端记事贴使用

最后一次自动
保存的日期

最后编辑于: 2018-09-08 18:31:06

清空

单独保存

在此单独保存

点击【单独保存】可以把文字单独保存到此处。

最后编辑于: 2018-08-24 15:29:18

基于electron(nodejs)编写的简单文本编辑器, 保存功能需要链接网络, 文本将自动保存至服务器。访问
web版:http://azusebox.moe/editor

最后编辑于: 2018-09-05 11:32:43

第三部分 实践总结

3.1 所做的工作

课设中所做的两个项目都是使用 html5+js 的方式编写的，我之所以会选择这样的方式一是因为助教放宽了对语言的要求，允许我们使用任何语言编写，二是我对 html 和 js 比较熟悉，此前参与过几个项目都是使用 js 编写的。使用一套 css 样式集我很容易就可以创建一个比较好看的图形画界面，同时借助这次课程设计的机会，我研究了一下 nodejs 的使用，我一直想知道如何用 html+js 编写 windows 可执行的程序而不是只能在浏览器中运行，我发现 electron 就是我需要的东西，许多我正在用的程序比如网易云音乐、微信开发者工具都是使用 electron 编写的。在我学习 electron 的途中我还找到 windowscmd 下特别好用的包管理软件 chocolate，简直是相见很晚。总之经过一番波折，我顺利使用 choco 配置好我的环境并使用 electron 编译出了可执行文件，相比使用 qt 或者 c#编写开发周期更短，更方便使用。

3.2 总结和收获

这次课设我最大的收获就是入门了 electron 的使用，学习 ES6 和 nodejs 的路还有很长，同学们很多都选择了 qt 或者 c#开发，选择 js 基本上就等于选择了前端道路，虽然我喜欢设计漂亮的软件的界面和思考怎么才能让用户更方便的使用我的软件，但我觉得学习一些比较难上手的 c 有关的知识也是十分有必要的。对于堆这个数据结构的学习也让我意识到了还是有很多很重要很基础的数据结构是我并没有完全掌握，作为一个程序员怎么能连堆是什么都不知道，而我这次也只做了堆有关的学习，需要我掌握的数据结构知识还有很多，因此我还需要继续努力，才能在未来的竞争中脱颖而出。

第四部分 参考文献

[1] 维基百科-堆排序 <https://zh.wikipedia.org/wiki/%E5%A0%86%E6%8E%92%E5%BA%8F>

[2] 维基百科-堆

[https://zh.wikipedia.org/wiki/%E5%A0%86_\(%E6%95%Bo%E6%8D%AE%E7%BB%93%E6%9E%84\)](https://zh.wikipedia.org/wiki/%E5%A0%86_(%E6%95%Bo%E6%8D%AE%E7%BB%93%E6%9E%84))