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[0]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,0,i) |

| | | | |*调换0,i元素 |*

| | | | |---drawswap(arr,0,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 |*

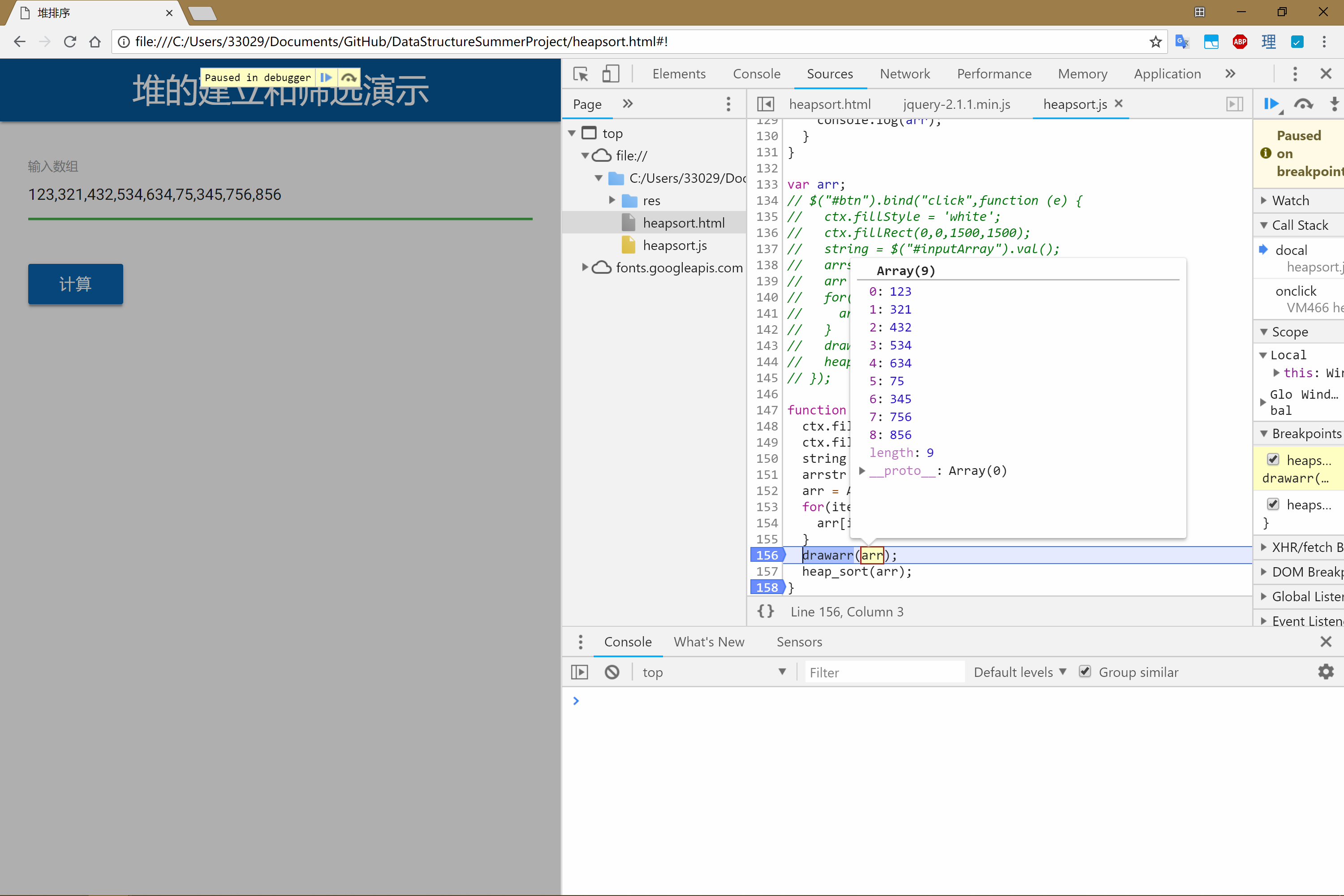
*| | | |i<0循环结束计算完成 >----------------------------------- |*

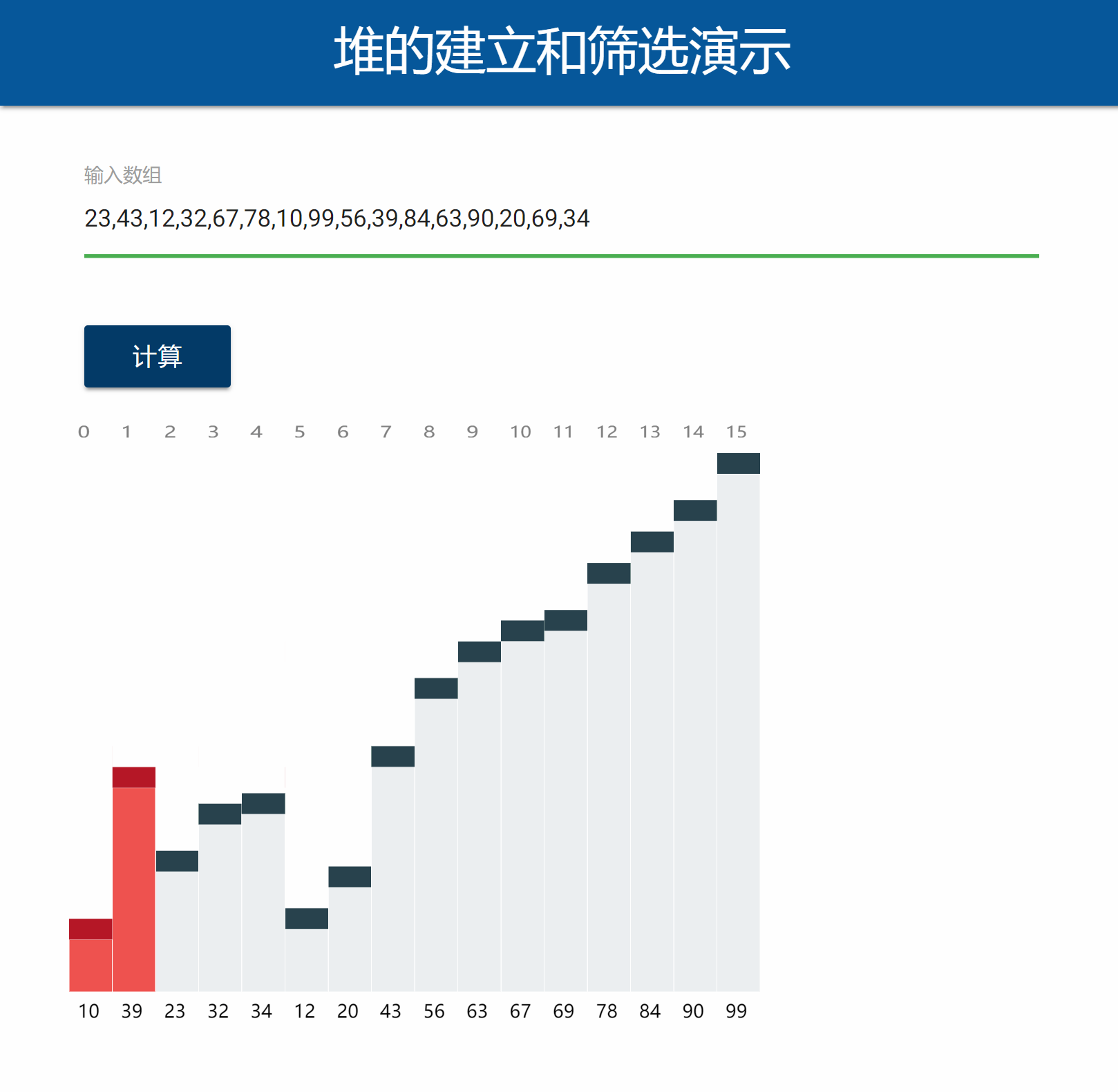
# 1.5 开发平台

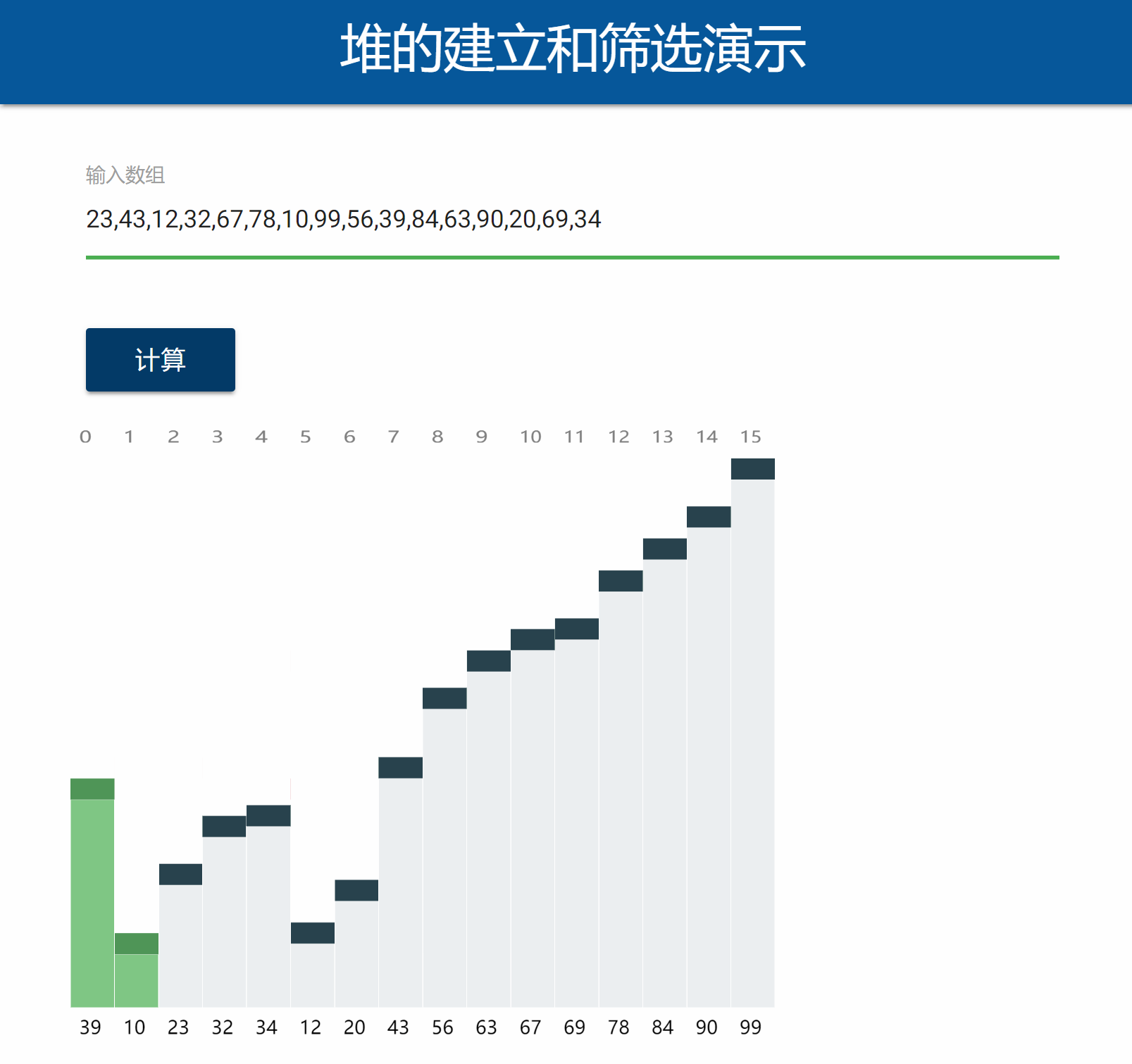
* HTML5
* Javascript ES6
* Jquery （js扩展库 https://jquery.com/）
* Materializecss(开源css样式集 <http://www.materializecss.cn/>)
* Exe文件使用nodejs与electron编译（https://www.electron.build/）
* 运行环境：支持html5 canvas的浏览器或任意windows电脑

# 1.6 系统的运行结果分析说明

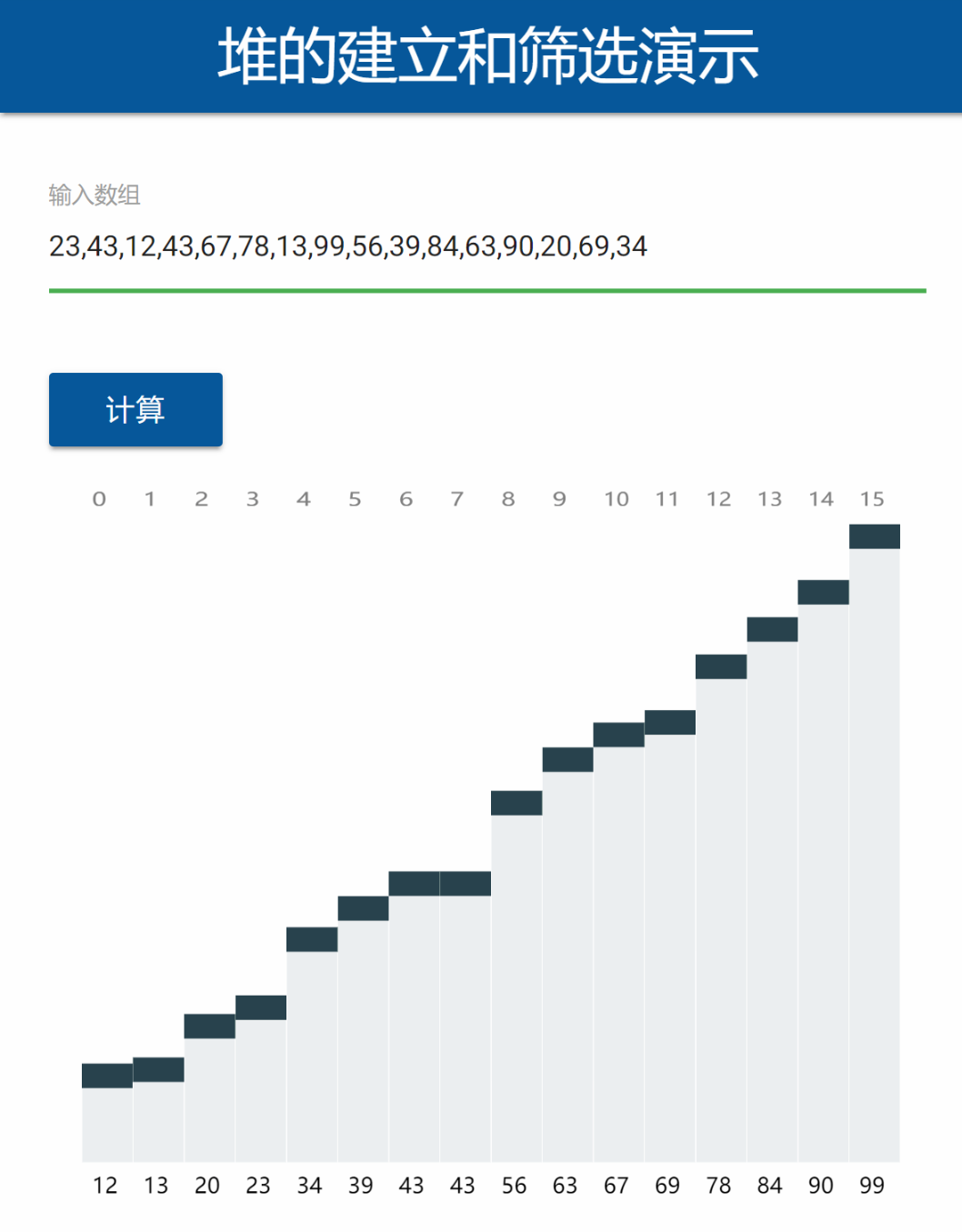
使用chrome开发者工具对代码进行调试



代码工作正常

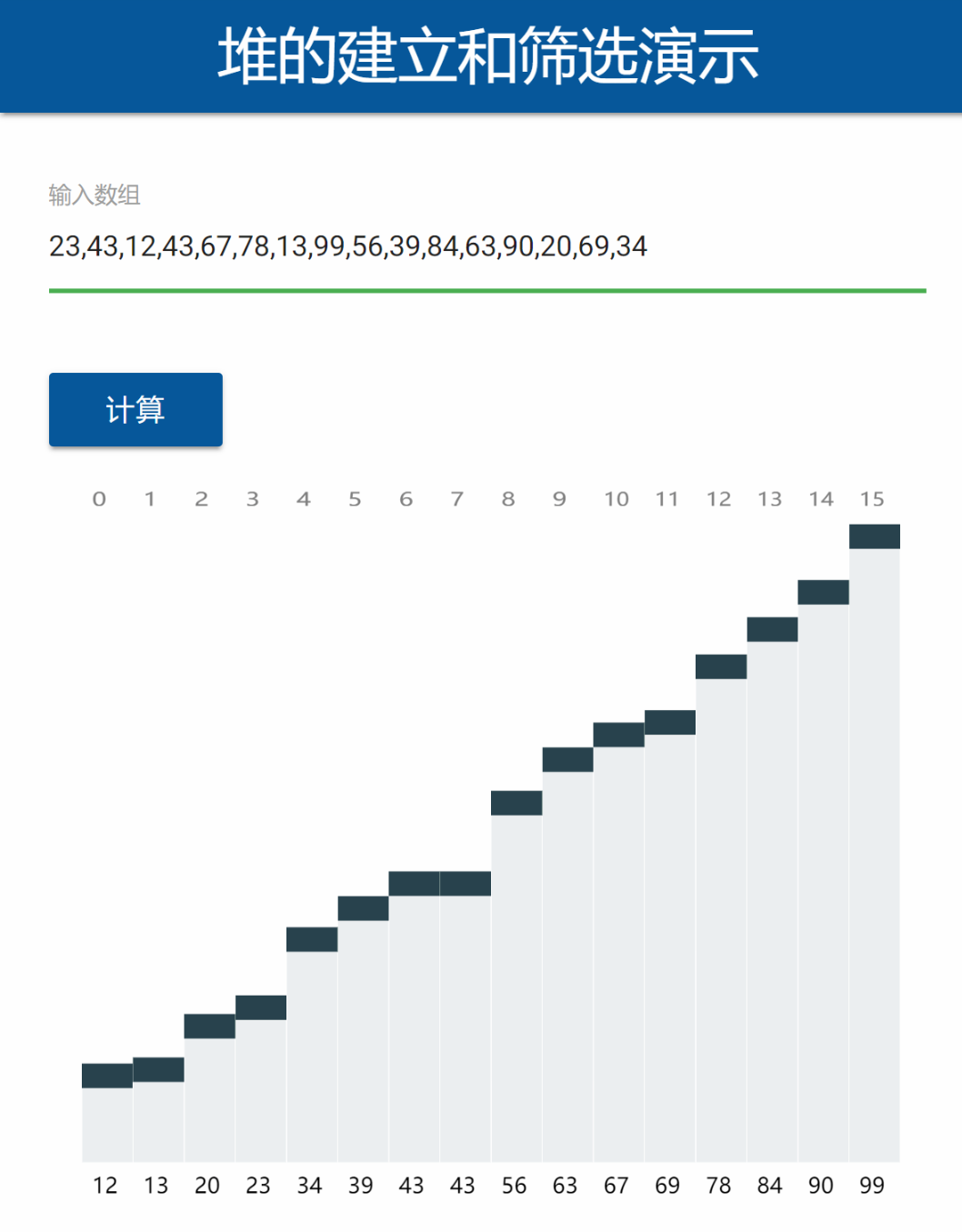


运行结果



# 1.7 操作说明

在此输入堆数组，以 , 间隔



计算结果会实时显示在此

点此开始计算

第二部分 综合应用部分设计说明