****

**数据可视化实验报告**



**学 院 智能与计算学部**

**专 业 软件工程**

**年 级 2018**

**学 号 3018216031**

**姓 名 白文杰**

**2021年 4 月 19 日**

目录

[一、 实验要求 2](#_Toc69764077)

[1. 绘制出bar chart 2](#_Toc69764078)

[2. 实现四种对bar chart的操作，并以动画的方式呈现变化： 2](#_Toc69764079)

[二、实验步骤 2](#_Toc69764080)

[1. 配置d3环境 2](#_Toc69764081)

[2. 初始化组件，包括button和svg 2](#_Toc69764082)

[3. 使用javascript和d3动态加载网页的初始数据 3](#_Toc69764083)

[4. 定义更新数据的操作，其中包括了bar和对应的值 4](#_Toc69764084)

[5. 实现四种操作 5](#_Toc69764085)

[三、结果分析 6](#_Toc69764086)

[四、源码 6](#_Toc69764087)

[index.html 6](#_Toc69764088)

[myd3.js 7](#_Toc69764089)

# 实验要求

## 绘制出bar chart

## 实现四种对bar chart的操作，并以动画的方式呈现变化：

* Sort：按照绑定的值的从小到大排序
* Shuffle：打乱
* Delete last one：删除最下方的bar
* Add random one：在最下方添加一个随机大小的bar

# 二、实验步骤

## 1. 配置d3环境

|  |
| --- |
| <script type=**"text/javascript"** src=**"js/d3.min.js"**></script> |

## 初始化组件，包括button和svg

|  |
| --- |
| <div id=**"buttons"**>  <button type=**"button"** onclick=**"sort()"**>**Sort**</button>  <button type=**"button"** onclick=**"shuffle()"**>**Shuffle**</button>  <button type=**"button"** onclick=**"delete\_last\_one()"**>**Delete last one**</button>  <button type=**"button"** onclick=**"add\_one()"**>**Add random one**</button>  </div>  <svg id=**"svg\_part"**></svg> |

## 使用javascript和d3动态加载网页的初始数据

|  |
| --- |
| // 初始化  **function** init**(){**  rect\_height **=** **innerHeight/**data**.**length**-**gap**;**  i **=** 0  data**.**forEach**(**d **=>** **{**  i **+=** 1**;**  g**.**append**(**'rect'**)**  **.**attr**(**'width'**,** **(innerWidth/**d3**.**max**(**data**))\***d**)**  **.**attr**(**'height'**,** rect\_height**)**  **.**attr**(**'fill'**,** '#56B6C2'**)**  **.**attr**(**'y'**,** i **\*** **(**rect\_height **+** gap**))**    g**.**append**(**'text'**)**  **.**attr**(**'class'**,** 'values'**)**  **.**attr**(**'x'**,** **-**30**)**  **.**attr**(**'y'**,** i **\*** **(**rect\_height **+** gap**)+**rect\_height**/**1.5**)**  **.**attr**(**'fill'**,** 'black'**)**  **.text(**d**)**  **});**    g**.**append**(**'text'**).text(**'Bar Charts'**).**attr**(**'font-size'**,** '22px'**).**attr**(**'transform'**,** `translate(${innerWidth/2}, ${-36})`**).**attr**(**'text-anchor'**,** 'middle'**);**  g**.**append**(**'text'**).**attr**(**'style'**,** 'white-space:pre'**).text(**'姓名：白文杰 学号：3018216031 '**).**attr**(**'font'**,**'console'**).**attr**(**'font-size'**,** '16px'**).**attr**(**'transform'**,** `translate(${innerWidth/2+10}, ${-10})`**).**attr**(**'text-anchor'**,** 'middle'**);**    **}** |

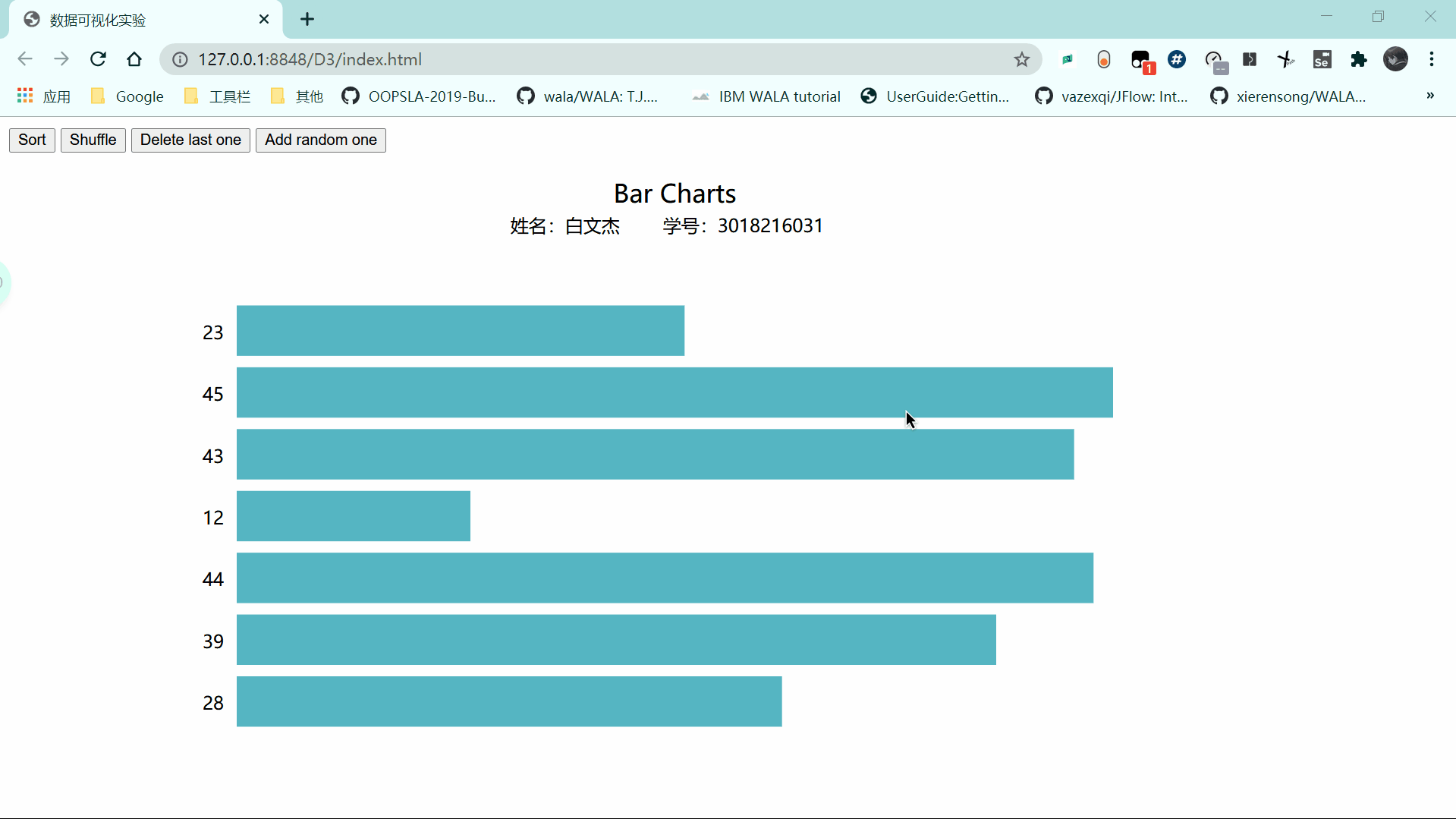
## 定义更新数据的操作，其中包括了bar和对应的值

|  |
| --- |
| **function** update\_data**(**data**){**  rect\_height **=** **innerHeight/**data**.**length**-**gap**;**  console**.**log**(**'现在的数组：'**+**data**);**  rects **=** g**.**selectAll**(**'rect'**).**data**(**data**);**  **text** **=** g**.**selectAll**(**'.values'**).**data**(**data**);**  update **=** rects**;**  enter **=** rects**.**enter**();**  exit **=** rects**.**exit**();**    // update.attr()    enter**.**append**(**'rect'**).**transition**()**  **.**attr**(**'width'**,** **function(**d**){**  **return** **(innerWidth/**d3**.**max**(**data**))\***d**;**  **})**  **.**attr**(**'height'**,** rect\_height**)**  **.**transition**()**  **.**duration**(**500**)**  **.**attr**(**'fill'**,** 'green'**)**  **.**attr**(**'y'**,** **function(**d**,** i**){**  **return** i **\*** **(**rect\_height **+** gap**);**  **})**  **.**duration**(**1360**);**    update**.**transition**()**  **.**attr**(**'width'**,** **function(**d**){**  **return** **(innerWidth/**d3**.**max**(**data**))\***d**;**  **})**  **.**attr**(**'height'**,** rect\_height**)**  **.**attr**(**'fill'**,** '#56B6C2'**)**  **.**attr**(**'text'**,** d **=>** d**)**  **.**attr**(**'y'**,** **function(**d**,** i**){**  **return** i **\*** **(**rect\_height **+** gap**);**  **})**    // .attr("fill","steelblue")  **.**duration**(**1360**);**    exit**.**transition**().**remove**().**duration**(**1360**\***0.3**);**    update **=** **text;**  enter **=** **text.**enter**();**  exit **=** **text.**exit**();**    enter**.**append**(**'text'**).**transition**()**  **.**attr**(**'class'**,** 'values'**)**  **.**attr**(**'x'**,** **-**30**)**  **.**attr**(**'y'**,** **function(**d**,** i**){return** i **\*** **(**rect\_height **+** gap**)+**rect\_height**/**1.5**})**  **.**attr**(**'fill'**,** '#56B6C2'**)**  **.text(function(**d**){return** d**})**  **.**duration**(**1360**\***1**);**    update**.**transition**()**  **.**attr**(**'y'**,** **function(**d**,** i**){return** i **\*** **(**rect\_height **+** gap**)+**rect\_height**/**1.5**})**  **.**attr**(**'fill'**,** '#56B6C2'**)**  **.text(function(**d**){return** d**})**  **.**duration**(**1360**\***1.3**);**    exit**.**transition**().**remove**().**duration**(**1360**\***0.3**);**  **}** |

## 实现四种操作

|  |
| --- |
| // 排序  **function** sort**(){**  //比较函数，否则sort排序会出错  **var** compare **=** **function** **(**x**,** y**)** **{**  **if** **(**x **<** y**)** **{**  **return** **-**1**;**  **}** **else** **if** **(**x **>** y**)** **{**  **return** 1**;**  **}** **else** **{**  **return** 0**;**  **}**  **}**  data **=** data**.**sort**(**compare**);**  update\_data**(**data**);**  **}**  // 打乱  **function** shuffle**(){**  data **=** d3**.**shuffle**(**data**);**  update\_data**(**data**);**  **}**  //删除最后一个  **function** delete\_last\_one**(){**  data **=** data**.**slice**(**0**,** data**.**length**-**1**);**  update\_data**(**data**);**  **}**  // 随机添加一个  **function** add\_one**(){**  **var** add\_value **=** Math**.**floor**(**Math**.**random**()\***100**);**  data**.**push**(**add\_value**);**  update\_data**(**data**);**  **}** |

# 三、结果分析

实验效果如下：

# 四、源码

## index.html

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset=**"utf-8"** />  <title>**数据可视化实验**</title>  </head>  <body>  <div id=**"buttons"**>  <button type=**"button"** onclick=**"sort()"**>**Sort**</button>  <button type=**"button"** onclick=**"shuffle()"**>**Shuffle**</button>  <button type=**"button"** onclick=**"delete\_last\_one()"**>**Delete last one**</button>  <button type=**"button"** onclick=**"add\_one()"**>**Add random one**</button>  </div>  <svg id=**"svg\_part"**></svg>  </body>  <script type=**"text/javascript"** src=**"js/d3.min.js"**></script>  <script type=**"text/javascript"** src=**"js/myd3.js"**></script>  </html> |

## myd3.js

|  |
| --- |
| **const** svg\_width **=** 1000**;**  **const** svg\_height **=** 500**;**  **const** svg **=** d3**.select(**'#svg\_part'**)**  **.**style**(**'margin-top'**,** '20px'**)**  **.**attr**(**'width'**,** svg\_width**)**  **.**attr**(**'height'**,** svg\_height**);**    data **=** **[**23**,** 45**,** 43**,** 12**,** 44**,** 39**,** 28**]**    **const** margin **=** **{top:** 60**,** right**:** 30**,** bottom**:** 60**,** left**:** 200**};**  **const** **innerWidth** **=** svg\_width **-** margin**.**left **-** margin**.**right**;**  **const** **innerHeight** **=** svg\_height **-** margin**.top** **-** margin**.**bottom**;**  **const** g **=** svg**.**append**(**'g'**).**attr**(**'id'**,** 'group\_part'**).**attr**(**'transform'**,** `translate(${margin.left}, ${margin.top})`**);**  **const** gap **=** 10**;**  // 初始化  **function** init**(){**  rect\_height **=** **innerHeight/**data**.**length**-**gap**;**  i **=** 0  data**.**forEach**(**d **=>** **{**  i **+=** 1**;**  g**.**append**(**'rect'**)**  **.**attr**(**'width'**,** **(innerWidth/**d3**.**max**(**data**))\***d**)**  **.**attr**(**'height'**,** rect\_height**)**  **.**attr**(**'fill'**,** '#56B6C2'**)**  **.**attr**(**'y'**,** i **\*** **(**rect\_height **+** gap**))**    g**.**append**(**'text'**)**  **.**attr**(**'class'**,** 'values'**)**  **.**attr**(**'x'**,** **-**30**)**  **.**attr**(**'y'**,** i **\*** **(**rect\_height **+** gap**)+**rect\_height**/**1.5**)**  **.**attr**(**'fill'**,** 'black'**)**  **.text(**d**)**  **});**    g**.**append**(**'text'**).text(**'Bar Charts'**).**attr**(**'font-size'**,** '22px'**).**attr**(**'transform'**,** `translate(${innerWidth/2}, ${-36})`**).**attr**(**'text-anchor'**,** 'middle'**);**  g**.**append**(**'text'**).**attr**(**'style'**,** 'white-space:pre'**).text(**'姓名：白文杰 学号：3018216031 '**).**attr**(**'font'**,**'console'**).**attr**(**'font-size'**,** '16px'**).**attr**(**'transform'**,** `translate(${innerWidth/2+10}, ${-10})`**).**attr**(**'text-anchor'**,** 'middle'**);**    **}**  // 排序  **function** sort**(){**  //比较函数，否则sort排序会出错  **var** compare **=** **function** **(**x**,** y**)** **{**  **if** **(**x **<** y**)** **{**  **return** **-**1**;**  **}** **else** **if** **(**x **>** y**)** **{**  **return** 1**;**  **}** **else** **{**  **return** 0**;**  **}**  **}**  data **=** data**.**sort**(**compare**);**  update\_data**(**data**);**  **}**  // 打乱  **function** shuffle**(){**  data **=** d3**.**shuffle**(**data**);**  update\_data**(**data**);**  **}**  //删除最后一个  **function** delete\_last\_one**(){**  data **=** data**.**slice**(**0**,** data**.**length**-**1**);**  update\_data**(**data**);**  **}**  // 随机添加一个  **function** add\_one**(){**  **var** add\_value **=** Math**.**floor**(**Math**.**random**()\***100**);**  data**.**push**(**add\_value**);**  update\_data**(**data**);**  **}**  **function** update\_data**(**data**){**  rect\_height **=** **innerHeight/**data**.**length**-**gap**;**  console**.**log**(**'现在的数组：'**+**data**);**  rects **=** g**.**selectAll**(**'rect'**).**data**(**data**);**  **text** **=** g**.**selectAll**(**'.values'**).**data**(**data**);**  update **=** rects**;**  enter **=** rects**.**enter**();**  exit **=** rects**.**exit**();**    // update.attr()    enter**.**append**(**'rect'**).**transition**()**  **.**attr**(**'width'**,** **function(**d**){**  **return** **(innerWidth/**d3**.**max**(**data**))\***d**;**  **})**  **.**attr**(**'height'**,** rect\_height**)**  **.**transition**()**  **.**duration**(**500**)**  **.**attr**(**'fill'**,** 'green'**)**  **.**attr**(**'y'**,** **function(**d**,** i**){**  **return** i **\*** **(**rect\_height **+** gap**);**  **})**  **.**duration**(**1360**);**    update**.**transition**()**  **.**attr**(**'width'**,** **function(**d**){**  **return** **(innerWidth/**d3**.**max**(**data**))\***d**;**  **})**  **.**attr**(**'height'**,** rect\_height**)**  **.**attr**(**'fill'**,** '#56B6C2'**)**  **.**attr**(**'text'**,** d **=>** d**)**  **.**attr**(**'y'**,** **function(**d**,** i**){**  **return** i **\*** **(**rect\_height **+** gap**);**  **})**    // .attr("fill","steelblue")  **.**duration**(**1360**);**    exit**.**transition**().**remove**().**duration**(**1360**\***0.3**);**    update **=** **text;**  enter **=** **text.**enter**();**  exit **=** **text.**exit**();**    enter**.**append**(**'text'**).**transition**()**  **.**attr**(**'class'**,** 'values'**)**  **.**attr**(**'x'**,** **-**30**)**  **.**attr**(**'y'**,** **function(**d**,** i**){return** i **\*** **(**rect\_height **+** gap**)+**rect\_height**/**1.5**})**  **.**attr**(**'fill'**,** '#56B6C2'**)**  **.text(function(**d**){return** d**})**  **.**duration**(**1360**\***1**);**    update**.**transition**()**  **.**attr**(**'y'**,** **function(**d**,** i**){return** i **\*** **(**rect\_height **+** gap**)+**rect\_height**/**1.5**})**  **.**attr**(**'fill'**,** '#56B6C2'**)**  **.text(function(**d**){return** d**})**  **.**duration**(**1360**\***1.3**);**    exit**.**transition**().**remove**().**duration**(**1360**\***0.3**);**  **}**  init**();** |