

Lab 6_2.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="description" content="Data Visualization"/>
6      <meta name="keywords" content="HTML, CSS, D3"/>
7      <meta name="author" content="Dai Vy"/>
8      <title>Task 6.2 D3 Interactivity - Sort</title>
9
10     <script src="https://d3js.org/d3.v7.min.js"></script>
11     <style>
12
13     </style>
14 </head>
15 <body>
16
17     <button id="add">Add</button>
18     <button id="remove">Remove</button>
19     <button id="sort">Sort</button>
20     <h1>LAB 6.2 D3 Interactivity - Sort</h1>
21
22     <script>
23         var w = 500;
24         var h = 100;
25         var maxValue = 25;
26         var dataset = [24, 10, 29, 19, 8, 15, 20, 12, 9, 6, 21, 28];
27         var duration = 2000;
28
29         // Use scaleBand() to create an ordinal scaleable x-axis based on the range of the
data set.
30         var xScale = d3.scaleBand()
31             .domain(d3.range(dataset.length))
32             .rangeRound([0, w])
33             .paddingInner(0.05);
34
35         // Use scaleLinear() to create a linear scaleable y-axis based on the range of the
data set.
36         var yScale = d3.scaleLinear()
37             .domain([0, d3.max(dataset)])
38             .range([0, h]);
39
40         var svg = d3.select("body")
41             .append("svg")
42             .attr("width", w)
43             .attr("height", h);
44
45         // Draw initial bars
46         svg.selectAll("rect")
```

```
47     .data(dataset)
48     .enter()
49     .append("rect")
50     .attr("x", function(d, i) {
51         return xScale(i);
52     })
53     .attr("y", function(d) {
54         return h - yScale(d);
55     })
56     .attr("width", xScale.bandwidth())
57     .attr("height", function(d) {
58         return yScale(d);
59     })
60     .attr("fill", "steelblue")
61     // Mouseover effect
62     .on("mouseover", function(event, d) {
63         d3.select(this)
64             .transition()
65             .duration(200)
66             .attr("fill", "orange");
67         // SVG tooltip
68         svg.append("text")
69             .attr("class", "svg-tooltip")
70             .attr("x", +d3.select(this).attr("x") + xScale.bandwidth()/2)
71             .attr("y", +d3.select(this).attr("y") - 8)
72             .attr("text-anchor", "middle")
73             .attr("font-size", "14px")
74             .attr("font-weight", "bold")
75             .attr("fill", "black")
76             .text(d);
77     })
78     .on("mouseout", function(event, d) {
79         d3.select(this)
80             .transition()
81             .duration(200)
82             .attr("fill", "steelblue");
83         svg.selectAll(".svg-tooltip").remove();
84     });
85
86     // Draw initial labels
87     svg.selectAll("text")
88         .data(dataset)
89         .enter()
90         .append("text")
91         .text(function(d) {
92             return d; // This displays the number on each bar
93         })
94         .attr("x", function(d, i) {
95             return xScale(i) + xScale.bandwidth() / 2;
96         })
```

```
97     .attr("y", function(d) {
98         return h - yScale(d) + 14;
99     })
100     .attr("text-anchor", "middle")
101     .attr("fill", "white")
102     .attr("font-size", "12px");
103
104 // Add bar function
105 function addBar() {
106     var newValue = Math.floor(Math.random() * maxValue);
107     dataset.push(newValue);
108     xScale.domain(d3.range(dataset.length));
109     yScale.domain([0, d3.max(dataset)]);
110     var delayStep = duration / dataset.length;
111
112     // Bars
113     var bars = svg.selectAll("rect")
114         .data(dataset, function(d, i) { return i; });
115
116     // ENTER new bar
117     var barsEnter = bars.enter()
118         .append("rect")
119         .attr("x", w) // start off right edge
120         .attr("y", function(d) { return h - yScale(d); })
121         .attr("width", xScale.bandwidth())
122         .attr("height", function(d) { return yScale(d); })
123         .attr("fill", "steelblue");
124     // Mouseover effect for new bars
125     .on("mouseover", function(event, d) {
126         d3.select(this)
127             .transition()
128             .duration(200)
129             .attr("fill", "orange");
130         svg.append("text")
131             .attr("class", "svg-tooltip")
132             .attr("x", +d3.select(this).attr("x") + xScale.bandwidth()/2)
133             .attr("y", +d3.select(this).attr("y") - 8)
134             .attr("text-anchor", "middle")
135             .attr("font-size", "14px")
136             .attr("font-weight", "bold")
137             .attr("fill", "black")
138             .text(d);
139     })
140     .on("mouseout", function(event, d) {
141         d3.select(this)
142             .transition()
143             .duration(200)
144             .attr("fill", "steelblue");
145         svg.selectAll(".svg-tooltip").remove();
146     });
```

```
147
148 // MERGE and transition all bars
149 bars.merge(barsEnter)
150   .transition()
151   .duration(duration)
152   .delay(function(d, i) { return i * delayStep; })
153   .attr("x", function(d, i) { return xScale(i); })
154   .attr("y", function(d) { return h - yScale(d); })
155   .attr("width", xScale.bandwidth())
156   .attr("height", function(d) { return yScale(d); });
157
158 // Labels
159 var labels = svg.selectAll("text")
160   .data(dataset, function(d, i) { return i; });
161
162 var labelsEnter = labels.enter()
163   .append("text")
164   .text(function(d) { return d; })
165   .attr("x", w + xScale.bandwidth() / 2)
166   .attr("y", function(d) { return h - yScale(d) + 14; })
167   .attr("text-anchor", "middle")
168   .attr("fill", "white")
169   .attr("font-size", "12px");
170
171 labels.merge(labelsEnter)
172   .transition()
173   .duration(duration)
174   .delay(function(d, i) { return i * delayStep; })
175   .text(function(d) { return d; })
176   .attr("x", function(d, i) { return xScale(i) + xScale.bandwidth() / 2; })
177   .attr("y", function(d) { return h - yScale(d) + 14; });
178 }
179
180 // Remove bar function
181 function removeBar() {
182   if (dataset.length === 0) return;
183   dataset.shift();
184   xScale.domain(d3.range(dataset.length));
185   yScale.domain([0, d3.max(dataset)]);
186   var delayStep = duration / (dataset.length || 1);
187
188 // Bars
189 var bars = svg.selectAll("rect")
190   .data(dataset, function(d, i) { return i; });
191
192 // EXIT first bar
193 bars.exit()
194   .transition()
195   .duration(duration)
196   .attr("x", w)
```

```
197         .remove();
198
199     // UPDATE remaining bars
200     bars.transition()
201         .duration(duration)
202         .delay(function(d, i) { return i * delayStep; })
203         .attr("x", function(d, i) { return xScale(i); })
204         .attr("y", function(d) { return h - yScale(d); })
205         .attr("width", xScale.bandwidth())
206         .attr("height", function(d) { return yScale(d); });
207
208     // Re-apply mouseover/mouseout to updated bars
209     bars.on("mouseover", function(event, d) {
210         d3.select(this)
211             .transition()
212             .duration(200)
213             .attr("fill", "orange");
214         svg.append("text")
215             .attr("class", "svg-tooltip")
216             .attr("x", +d3.select(this).attr("x") + xScale.bandwidth()/2)
217             .attr("y", +d3.select(this).attr("y") - 8)
218             .attr("text-anchor", "middle")
219             .attr("font-size", "14px")
220             .attr("font-weight", "bold")
221             .attr("fill", "black")
222             .text(d);
223     })
224     .on("mouseout", function(event, d) {
225         d3.select(this)
226             .transition()
227             .duration(200)
228             .attr("fill", "steelblue");
229         svg.selectAll(".svg-tooltip").remove();
230     });
231
232     // Labels
233     var labels = svg.selectAll("text")
234         .data(dataset, function(d, i) { return i; });
235
236     labels.exit()
237         .transition()
238         .duration(duration)
239         .attr("x", w)
240         .remove();
241
242     labels.transition()
243         .duration(duration)
244         .delay(function(d, i) { return i * delayStep; })
245         .text(function(d) { return d; })
246         .attr("x", function(d, i) { return xScale(i) + xScale.bandwidth() / 2; })
```

```

247         .attr("y", function(d) { return h - yScale(d) + 14; });
248     }
249
250     // Sort function
251     var sortAscending = true;
252     function sortBars() {
253         // Create an array of objects with value and original index
254         var indexedData = dataset.map(function(d, i) { return {value: d, index: i}; });
255         // Sort the array by value
256         indexedData.sort(function(a, b) {
257             return sortAscending ? d3.ascending(a.value, b.value) :
d3.descending(a.value, b.value);
258         });
259         // Update xScale domain to new order (array of original indices in sorted order)
260         xScale.domain(indexedData.map(function(d) { return d.index; }));
261         // Transition bars to new x positions
262         svg.selectAll("rect")
263             .transition()
264             .duration(duration)
265             .attr("x", function(d, i) {
266                 // Find the new x position for this bar's index
267                 var sortedIndex = indexedData.findIndex(function(obj) { return obj.index
=== i; });
268                 return xScale(i = indexedData[sortedIndex].index);
269             });
270         // Transition labels to new x positions
271         svg.selectAll("text")
272             .transition()
273             .duration(duration)
274             .attr("x", function(d, i) {
275                 var sortedIndex = indexedData.findIndex(function(obj) { return obj.index
=== i; });
276                 return xScale(i = indexedData[sortedIndex].index) + xScale.bandwidth() /
2;
277             });
278         // Toggle sort order for next click
279         sortAscending = !sortAscending;
280     }
281     d3.select("#sort").on("click", sortBars);
282
283     // Button event listeners
284     d3.select("#add").on("click", addBar);
285     d3.select("#remove").on("click", removeBar);
286 </script>
287
288 <br>
289
290 <footer style="color: grey">COS30045 Data Visualization<br>
291     Dai Vy
292 </footer>
293 </body>

```

294 | </html>