**Billiard Trees**

Link to GitHub repository: <https://github.com/kpsokol1/Team_4_Project_2>

Contributors: Kyle Sokol, Sungmin Kim, Chad Perry, Dixitha Korvi, Ben Fioresi, Tyler McWilliams, and Mark Chapman

Description: Billiard Trees is a browser-based animation program written in JavaScript and HTML. Our project is inspired by billiard balls which will hopefully make learning data structures more fun. We chose to use a browser-based implementation as it is the most accessible and no additional compilers or packages are needed apart from a supported browser.

Languages Used: JavaScript, HTML

Data Structures Implemented: Binomial Heap, Red-Black Tree, and B-Tree

How to Use:

1. To open up the project click on Project2.html in the root directory and open it in a browser.
2. Scroll to which tree structure you would like to animate.
3. Each data structure implements the following four operations which can be selected from the drop-down menu: Insert, Delete, Find, and Extract Min.
4. If you are running the Insert, Delete, or Find operation please enter a number (negatives or floating points are OK) in the text box to the right of drop-down menu.
5. Click submit to run the animation

\*Note: Duplicate keys are not supported

\*Note: B-Trees currently just support 2-4 trees (t = 2)

\*Note: There could be some glitches attempting to delete a node that doesn’t exist from a B-Tree

Pausing The Animations:

* The pause button can be pressed at any time to pause the animation, and the run button will resume the animation.

Playback History:

* After the animation is completed, you can drag to playback slider bar to the left or right to see past animations.
* You can also use the previous and next animation buttons to step through the animations.

Adjusting the Speed:

* The speed slider bar can be adjusted to the left or right to slow down or speed up the animations.

Supported Browsers:

* We have primarily tested on Chromium-based browsers such as Google Chrome and Microsoft Edge, but the project should also work with Mozilla Firefox.

File Structure:

* /res: contains any images and graphics used in the project
* /src: all source code
  + /BHeap: Source code for the Binomial Heap
    - BHeap.js: Binomial Heap data structure
    - BHeapAnimations.js: Animations for the Binomial Heap
  + /B-Tree: Source code for the B-Tree
    - B-Tree.js: B-Tree data structure
    - B-TreeAnimations.js: Animations for the B-Tree
  + /RBTree: Source code for the Red Black
    - RBTree.js: Red Black Tree data structure
    - BinaryTreeAnimations.js: Animations for the Red Black Tree
  + Animation.js: Class that handles the play, pause, playback, and timing of all animations
  + Common.js: Common library functions used throughout the implementation
  + Controller.js: Sets up the canvas and converts user input (numbers and button presses) into calls to the backend
  + TreeAnimations.js: Common animations used throughout the implementation
* Project2.html: File that visualizes all the animations in the browser
* Team4.pptx: Our class presentation