



Spice Debugger – Team Worm

Russell Johnston, Tarik Courdy, Elliot Hatch, Samuel Davidson

Visualization

Spice is an experimental debugger that visualizes function behavior across time. Instead of stepping through a program line-by-line, Spice captures its full behavior in an explorable form.

```

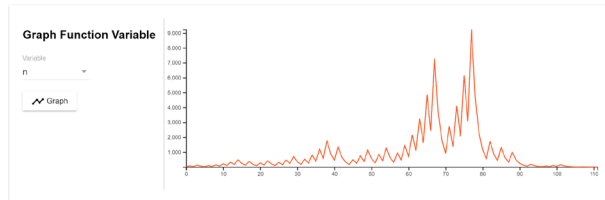
5 int binary_search(int x, int len, int arr[]) {
6     int min = 0;
7     int max = len - 1;
8
9     while (max >= min) {
10        int i = (min + max) / 2;
11        if (arr[i] == x) {
12            return i;
13        } else if (arr[i] > x) {
14            max = i - 1;
15        } else {
16            min = i + 1;
17        }
18    }
19    return -1;
20 }
21

```

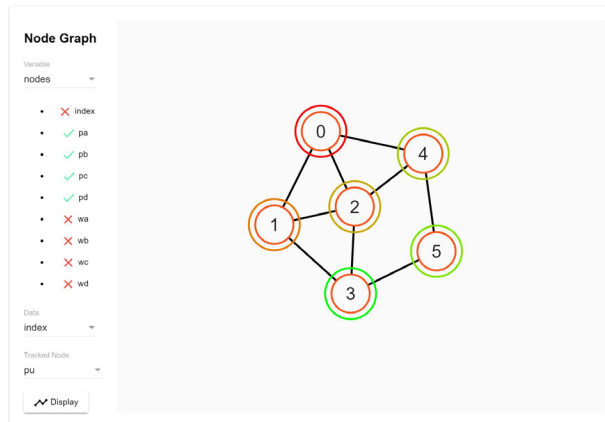
	i = 49	i = 74	i = 61	i = 67	i = 70	i = 68	i = 69
	max = 73				max = 69		returned 69
	min = 50	min = 62	min = 68		min = 69		

Control flow is displayed as a table, with loop iterations represented as columns. Loops can be expanded and collapsed, nested, and scrolled.

As function arguments are modified, the program is rerun and the visualization is updated.

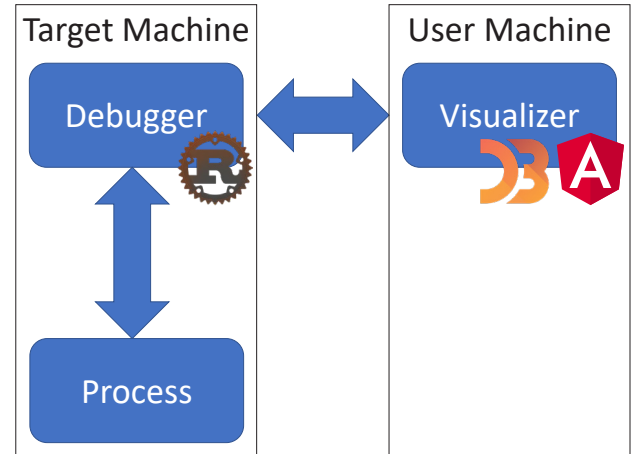


Scalar values are charted over the course of the execution of a program, exposing unexpected behavior and opportunities for optimization.



Complex data structures are mapped to visual representations that capture their traversal or mutation.

Architecture



Debugger

The debugger extracts type and symbol information from the target process, controls its execution, and traces the control and data flow of functions.

Visualizer

The visualizer collects trace data and provides visualizations of a function's behavior.