Conjure

Phil Chen & Grace Kwan
The Problem(s)

- Reverse a linked list.
- Traverse a tree iteratively.
- Determine whether an integer is a palindrome.
- Find the median of two sorted arrays.
- Reverse the order of words in a sentence.
- Find a cycle in a linked list.
- And so on…
The Solution

- Step-by-step code visualizer
- Custom API for JS data structures
- Question examples
Prior Work
Add Binary

Given two binary strings, return their sum (also a binary string).

For example,

\[a = "11"\]
\[b = "1"\]

Return "100".

» Solve this problem
Breadth-First Search

Start Vertex: □ Run BFS □ New Graph
○ Directed Graph ○ Small Graph ○ Logical Representation
○ Undirected Graph ○ Large Graph ○ Adjacency List Representation
○ Adjacency Matrix Representation

BFS Queue

<table>
<thead>
<tr>
<th>Parent</th>
<th>Visited</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

Animation Completed

Animation Speed

w: 1000 h: 500
Array

Stack

Queue

Binary Tree

Linked List

2D Arrays

Map

Keys: red, blue, green, black, pink

String: stanford

* Column on side for loose vars
Determine if a singly-linked list contains a cycle.

```javascript
function checkCycle(var head) {
    var run = head.next;
    while (run !== null) {
        if (head === run) {
            return true;
        } else {
            head = head.next;
            run = run.next.next;
        }
    }
    return false;
}
```
Completion Plan

API
- Phil: Graphs, Maps, Linked Lists, Heaps
- Grace: Arrays, Stacks, Queues, Strings

Web interface
- Phil: Code parsing
- Grace: UI design & implementation
Open Questions

- Are these representations natural?
- Other useful data structures?
- Interface design decisions?