

Google's interview process

For software engineers



Steps



Phone screen (I hadn't signed an NDA yet)

Write an iterator that combines items from of a list of iterators

Implement class `interlockingList` with member functions `next`, `pop`, `hasNext` such that

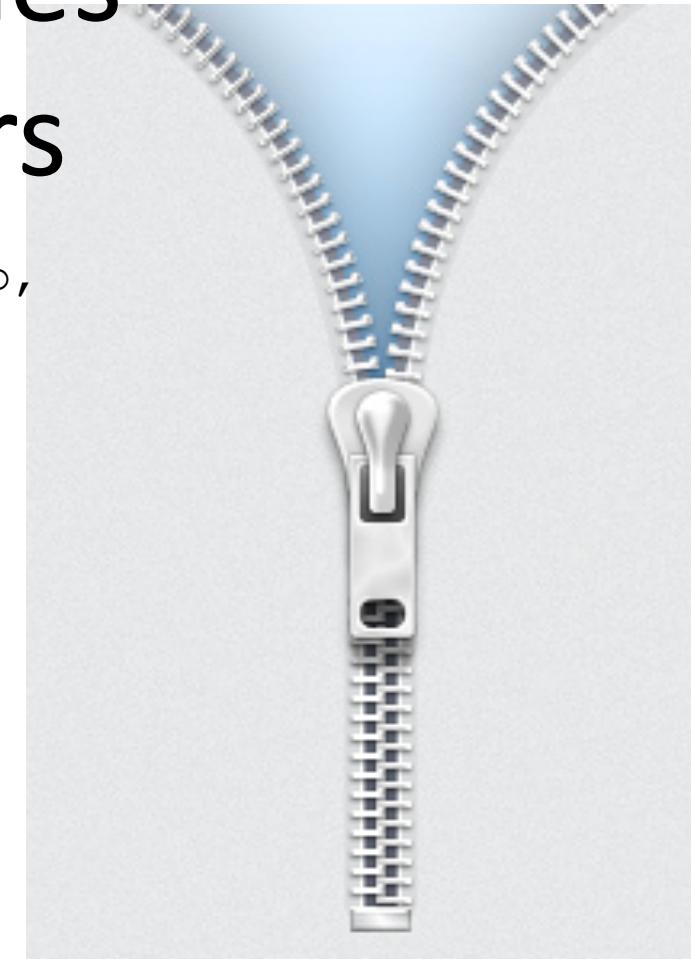
A = [1,1,1]

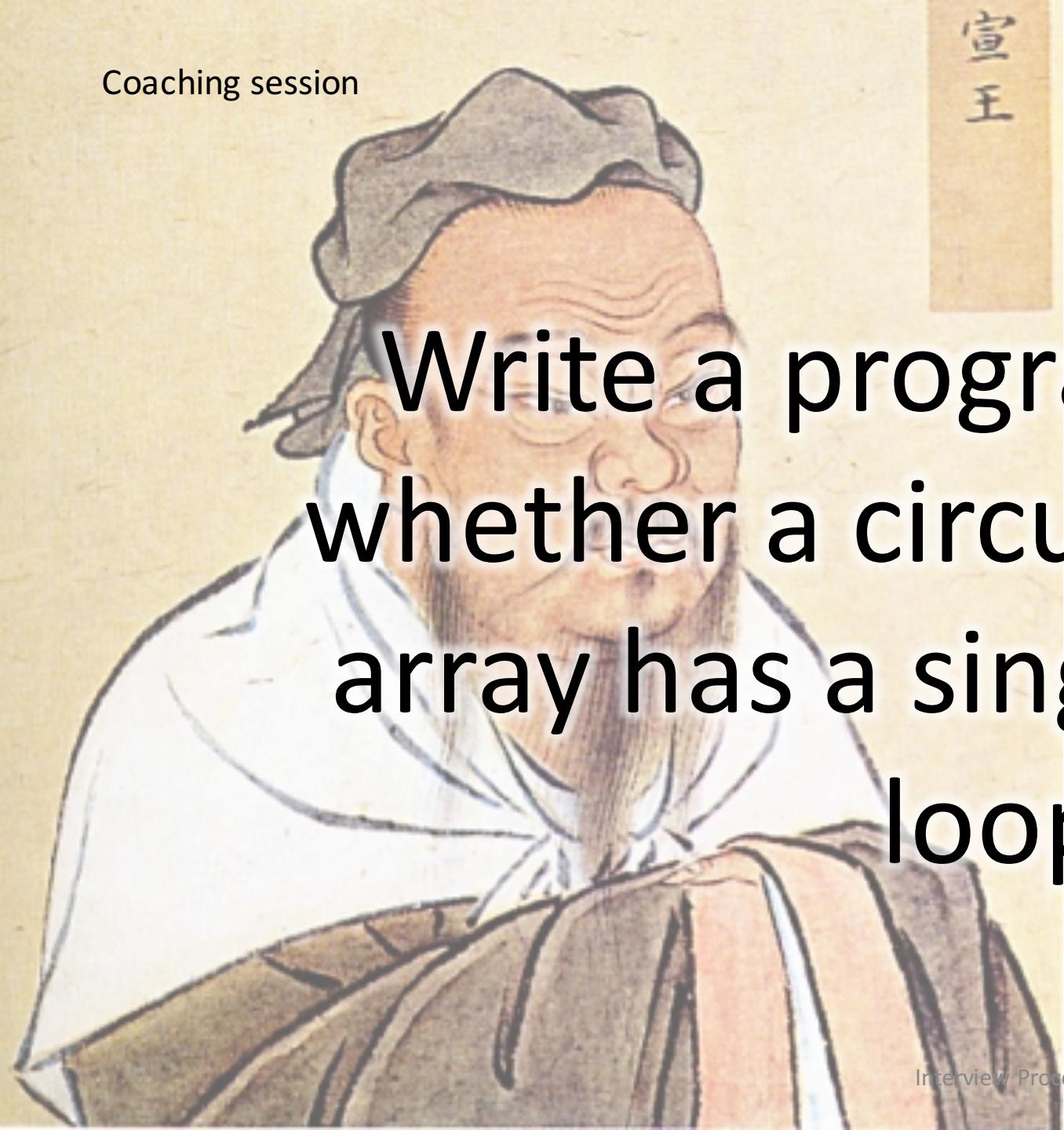
B = [2,2]

C = [3,3,3]

```
il = interlockingList([A,B])
while it.hasNext():
    print it.pop()
>> 1,2,1,2,1
```

```
il= interlockingList([A,B,C])
while it.hasNext():
    print it.pop()
>> 1,2,3,1,2,3,1,3
```





Write a program to check
whether a circular reference
array has a single complete
loop.

Computer science basics

- Big-O
 - Know how to estimate for hash tables, heaps, binary trees, linked lists, depth-first search, recursion
- Sorting
 - Know at least one $n \log(n)$ sorting algorithm (mergesort or quicksort)
- Hash tables
 - Know what it is and how to implement it
- Trees
 - Binary, n-ary, search trees
 - Tree size/complexity
 - Know at least one kind of balanced tree: AVL, red-black, splay
- Graph theory
 - Representations: pointer, matrix, adjacency list
 - Breadth first search vs depth first search (also inorder, postorder preorder)
 - Advanced algorithms:
 - MST: Krushkal, Prim
 - Shortest path: Dijkstra/A*

On-site

- 5 interviews + lunch “interview”
- Coding on a white board
- Interview details are NDAd (sorry)



Websites

Long blog post about getting hired at google

- <http://steve-yegge.blogspot.com/2008/03/get-that-job-at-google.html>

Coding practice

- Leetcode, codeforces, hackerrank

Information about algorithms and running time

- <http://bigocheatsheet.com/>

Summaries of topics in computer science and data science

- <https://www.topcoder.com/community/data-science/data-science-tutorials/>

Books

The Algorithm
Design Manual

- Steven Skiena

Cracking the
Coding Interview

- Gayle Laarkman
McDowell