

# Acing Your Technical Interview

Tony Ho Harvard 2014

What is the difference between a Stack and a Queue? How do you take care of conflicts in Hash Tables?



JANE STREET



facebook®

box

adaptec

AMDAHL  
a Fujitsu company

ATARI

NETFLIX

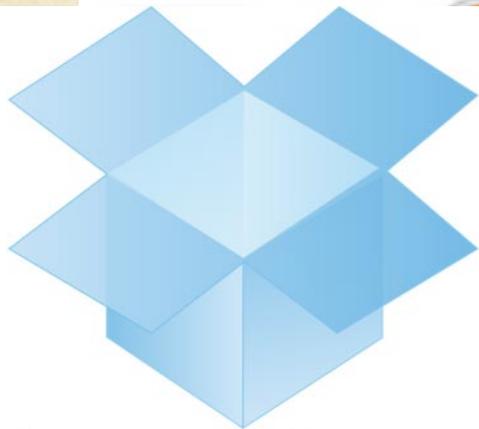
ASUS

Netscape



PayPa

Google



Dropbox

amazon.com

What happens when you visit a website? Describe everything you know.

# Background

- CS50 TF
- Google Summer 2010
- Facebook Summer 2011
- How do I prepare for my interviews?

Reverse, Trim, Remove  
Letters from, Check  
Palindrome on, Find  
Longest Palindrome in a  
String.

# How do they hire?

- Interviews!
  - Software Engineer
  - Product Manager
- Require Technical Background

Check if a number is a power of 2. In an array of size  $n-1$  with all but one number from  $1-n$ , find the missing number. Make the pow function.

# Why do they interview?

- Acquire the best talent for their company
  - Software Engineers - Code!
  - Product Manager - Innovate!
- The interview process then must be difficult and very selective

Given a linked list, extend the linked list with the current elements reversed. Reverse the linked list. Randomize the linked list.

# The Interview

- Series of 2 to 5 depending on the company
- 30-40 minutes each
- Two types:
  - Onsite : whiteboard coding
  - Over the Phone : Collab-edit coding
- 2-3 problems testing your computer science knowledge
- Yes, you may need to code even for product engineers

Write Binary Search,  
Traverse a Tree  
recursively. Given  
traversals (pre, in, post)  
determine the tree

# Types of Questions

1. Problem Solving / Algorithms
  - Find optimal solutions (Big-O)
  - Write the solution
2. Data Structures
  - Goes along with algorithms but they might directly as you questions about this
3. The internet
  - Information flow
  - Web development techniques
4. Designing
  - Database
  - Systems (Classes, modules, etc.)

Write Fibonacci  
recursively, iteratively.  
What are the differences?

# General Tips

- Know your recruits, don't be late, dress nicely
- Don't stop talking!
  - Fast responses
- Think out loud
- They want to see your code (even if it's pseudocode!)
- Ask for hints
  - They're nice people!

Create a deck of cards.  
Then implement shuffle.

# Data Structures

- Stacks
- Queues
- Linked-Lists
- Maps / Sets
- Hashtables
- Binary Trees / Trees
- Tries
- Red-Black Trees
- AVL Trees
- Graphs
- Heaps

Implement and give the  
Big-O for all data  
structures

# Algorithms

- Search
- Recursion
- String Manipulations
- Number Manipulations
- Flood Fill
- Shortest Path
- Greedy
- Network Flow
- Dynamic Programming
- Ad Hoc Problems
- CS124

Design an iterator for your data structures. Implement insert and delete as well.

# Preparation

- Problem solving. Competitive coding problems.
  - Do problems as if you are in an interview
  - Quiz your friends
1. [glassdoor.com](http://glassdoor.com), [Google](http://Google.com)
  2. [poj.org](http://poj.org)
  3. [projecteuler.net](http://projecteuler.net)
  4. [train.usaco.org](http://train.usaco.org)
  5. [topcoder.com](http://topcoder.com)
  6. [codeforces.com](http://codeforces.com)

Implement Merge Sort.  
Implement a Priority  
Queue.

# glassdoor.com

- People talk about their past interview experiences with companies
- Nowhere to turn in code
- People's solutions are mostly wrong or incomplete

How would you create one of the products that you use today? (Gmail, Facebook, iPhones)

# poj.org

- Great resource of problems
- Competitive programming style
- You can code up solutions and submit them
  - online judger
- Great way to practicing coding while looking at interesting problems that could potentially come up in an interview

Find the largest  
continuous subset of an  
array. {6,1,3,2,4} =>  
{1,2,3,4};  
{8, 9, 1, 2, 3} => {1,2,3}

# Competitive Programming

- File I/O. Scan file in. Print file out.
- Sample Problem:

```
#include<iostream>
using namespace std;
int main() {
    int a, b;
    cin << a << b;
    cout << a+b << endl;
    return 0;
}
```

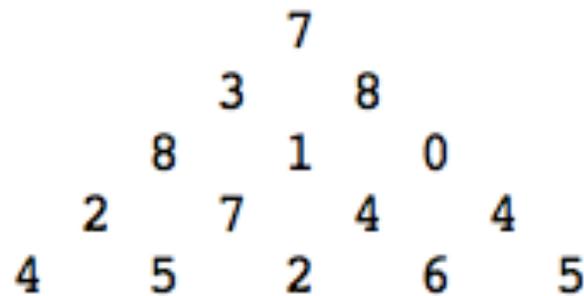
```
#include<stdio.h>

int main() {
    int a, b;
    scanf("%d %d", a, b)
    printf("%d", a+b)
    return 0;
}
```

Calculate  $a + b$  from an input file with  $a$  and  $b$  that are separated by a whitespace.

Questions?

# Case Study



(Figure 1)

Figure 1 shows a number triangle. Write a program that calculates the highest sum of numbers passed on a route that starts at the top and ends somewhere on the base. Each step can go either diagonally down to the left or diagonally down to the right.