

Zen and the art of Interviewing

Alex Maloney
ACM 2018

Interviews are about **coding**, right ...

... Interviews are about **communication!**

General Tips

Goal of an interview is to **Translate a problem into code**

Focus on being **clear** not clever

Clear code has a unified style and **well named functions and variables**

Ask clarifying questions.

The best interviews are full of mistakes

Pick a single language to do all your interviews in

Types of Interviews

**Timed
Challenge**

**Remote
Interview**

**Onsite
Interview**

Timed Challenge

Usually tests a very specific area of computer science. Closest to the typical “tricky” coding challenge.

Problem is well defined for you

Come up with your own test cases

Best Prep technique:



HackerRank

Remote Interview

Very challenging environment

Shared text editor and a phone call

Don't be afraid to ask questions

The goal is to demonstrate your coding process, not necessarily the end result

Talk through your solution as you develop it

Start with simple unoptimized case and develop from there

Best prep technique:



interviewing.io

Onsite Interview

Two main focuses ...

1. How well can you design a system
2. How well can you solve a small problem.

System Design Question

Architect a solution to a more complex problem

Organize data structures, classes, objects and functions

Example: Design a system to detect ships in a game of battleship

Focused Problem Question

Smaller focused problems - typically one function

Iterate multiple designs

Cover all edge cases - showing how you handle them within your code

Focus on code structure, have an intention behind each line and explain why you're writing it this way

Example : write a function that detects if a string is a palindrome

Battleship - Given a matrix representing a game of battleship, find the ships!

```
class Board {  
  
}  
  
class Board {  
    int[][] board_squares; //1 if ship 0 if empty - given info  
    Ship[] found_ships; //initially empty  
}  
  
class Ship {  
    int[] |  
  
class Board {  
    int[][] board_squares; //1 if ship 0 if empty - given info  
    Ship[] found_ships; //initially empty  
  
    boolean is_hit(int x, int y); //returns true if the given coordinate is a hit  
  
    boolean ship_already_found(Ship s); //true if this ship has already been hit and found  
  
    void record_ship(Ship s); //add ship to list of found ships  
}  
  
class Ship {  
    int[][] coords; //list of (x,y) pairs  
}
```

Battleship - Given a matrix representing a game of battleship, find the ships!

Board b

```
for y in b.board_squares {
  for x in b.board_squares {
    Ship current_ship = null;
    for y in b.board_squares {
      for x in b.board_squares {
        if (b.is_hit(x,y)) {
          if (current_ship == null) {
            current_ship = new Ship(x,y);
          }
          current_ship.add_hit(x,y);
        }
      }
    }
  }
}

Board b;
Ship current_ship = null;
for y in b.board_squares {
  for x in b.board_squares[x] {
    if (b.is_hit(x,y)) {
      if (current_ship == null) {
        current_ship = new Ship(x,y);
      }
      current_ship.add_hit(x,y);
    }
    //no hit but previous iteration was a hit
    else if (current_ship != null) {
      b.record_ship(current_ship);
      current_ship = null;
    }
  }
}
```

This has a fatal flaw ...

Write a function
that detects if a
string is not a
palindrome

? || /* */

Resources

<http://hackerrank.com/>

<https://interviewing.io/>

<https://www.reddit.com/r/dailyprogrammer/>

<https://news.ycombinator.com/>

<http://code.facebook.com/>

<https://developers.google.com/>