Technical Interview Questions

Interviewer Checklist

Make sure the interviewee does the following:

For EE:

- Explains all calculations (doesn't jump to a formula)
- Talks through their design process as they are writing/thinking
- Consider multiple options for certain functions (switch vs push button as actuator, for example)
- Explains why they chose one element vs. another

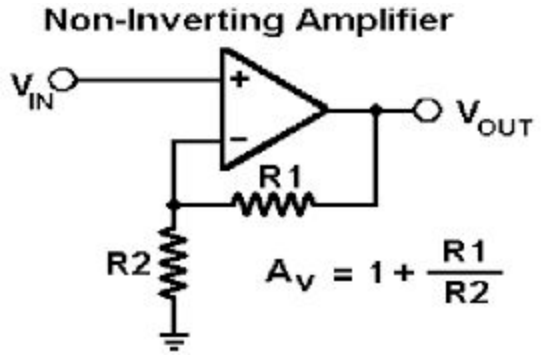
For CS:

- Is always talking about their thought process
- asks about/clarifies edge cases and terminology
- analyzes runtime and space complexity if relevant
- TESTS TESTS TESTS (multiple inputs that cover different cases)

https://docs.google.com/document/d/1AuFaVL5YbyRNea-zATi13S7STQlXqOVy8vJOV1uxnwk/edit

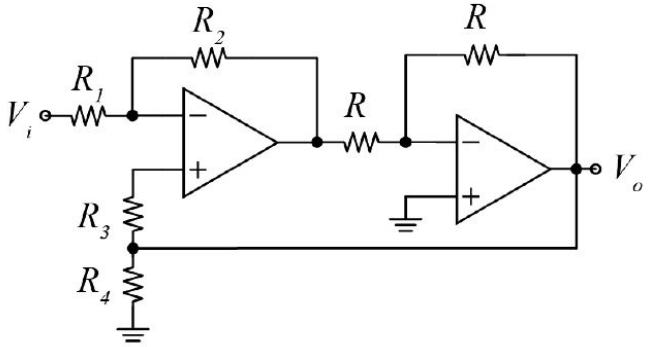
Electrical Engineering (see link above for more)

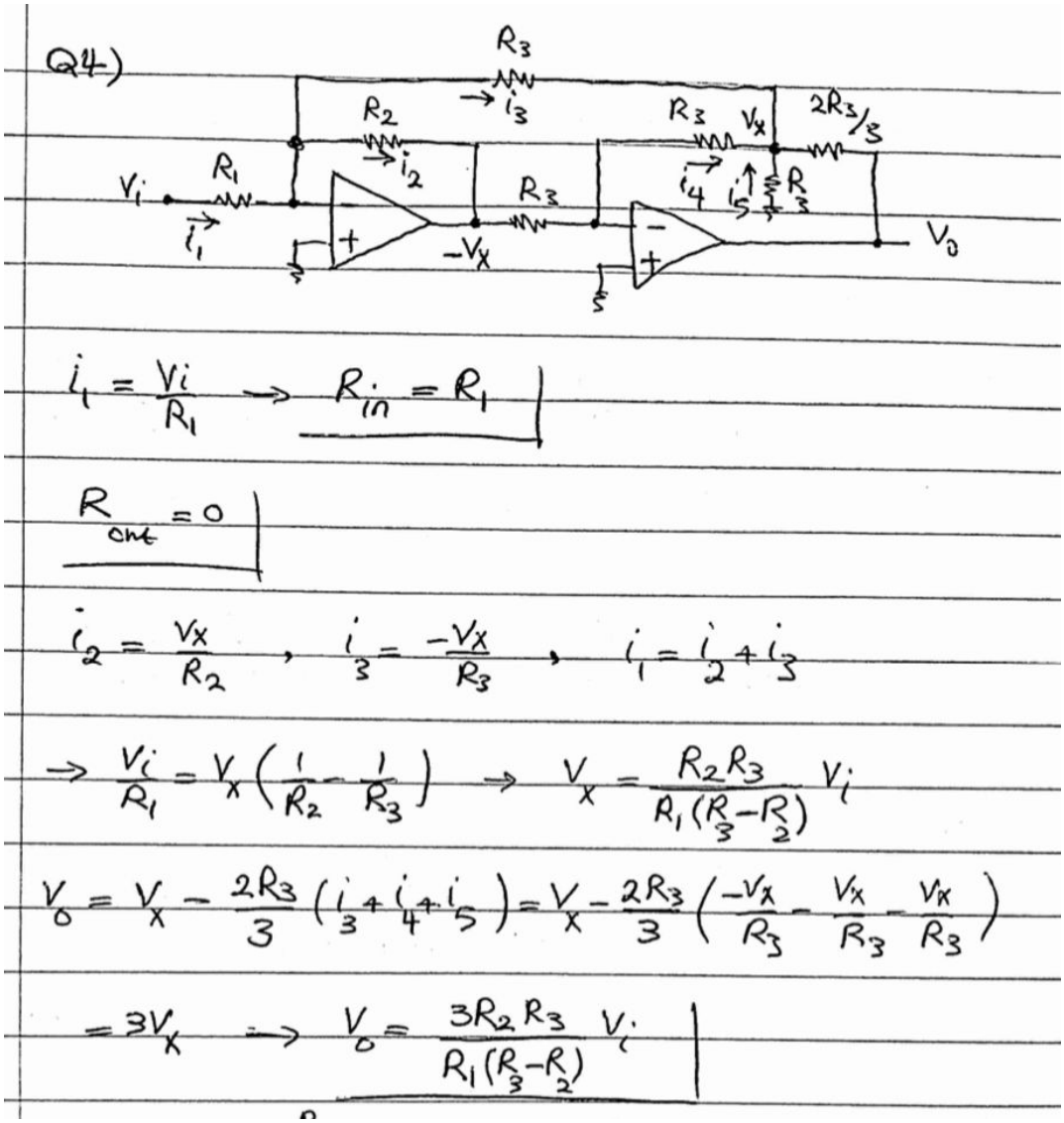
- *Draw out a non-inverting amplifier, but with some twist (terminals in nonstandard order, resistor in odd location, etc.* Analyze this circuit element and explain its function.



- Design a circuit such that when I press a button, 2 seconds later, an LED lights up.
    - Any functional design which they can explain/justify the behavior of.
- Derive an expression for an inverting amplifier with non-infinite open loop gain
    -
- Explain what a PN junction is and how it functions in 1) no bias 2) reverse bias 3) forward bias

Find an expression for V_o/V_i in terms of R, R_1, R_2, R_3, and R_4. The Opamps are ideal.

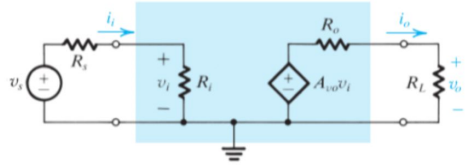




1.43 Consider the voltage-amplifier circuit model shown in Fig. 1.16(b), in which $A_{vo} = 100$ V/V under the following conditions:

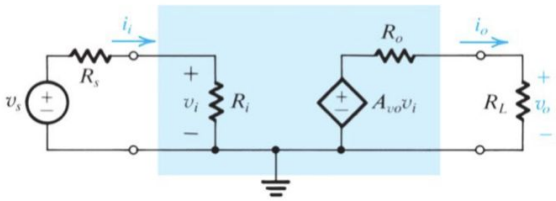
(a) $R_i = 10R_s$, $R_L = 10R_o$
(b) $R_i = R_s$, $R_L = R_o$
(c) $R_i = R_s/10$, $R_L = R_o/10$

Calculate the overall voltage gain v_o/v_s in each case, expressed both directly and in decibels.



3 [S & S P1.43]

$A_{vo} = 100\ V/V$



$$v_o = \frac{R_L}{R_o + R_L} A_{vo} v_i \quad (1)$$

$$v_i = \frac{R_i}{R_i + R_s} v_s \quad (2)$$

(2) into (1)

$$v_o = \left(\frac{R_L}{R_o + R_L}\right)\left(\frac{R_i}{R_i + R_s}\right) A_{vo} v_s$$

$$\frac{v_o}{v_s} = A_{vo}\left(\frac{R_i}{R_s + R_i}\right)\left(\frac{R_L}{R_o + R_L}\right)$$

(a) $R_i = 10R_s$, $R_L = 10R_o$

$$\frac{v_o}{v_s} = \left(100\frac{V}{V}\right)\left(\frac{10R_s}{11R_s}\right)\left(\frac{10R_o}{11R_o}\right) = 82.64\ V/V$$

$$\left|\frac{v_o}{v_s}\right|_{dB} = 20\log_{10}(82.64\ V/V) = 38.34\ dB$$

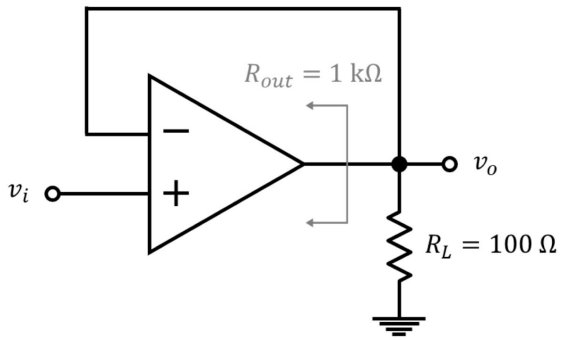
(b) $R_i = R_s$, $R_L = R_o$

$$\frac{v_o}{v_s} = \left(100\frac{V}{V}\right)\left(\frac{1R_s}{2R_s}\right)\left(\frac{1R_o}{2R_o}\right) = 25\ V/V = 27.96\ dB$$

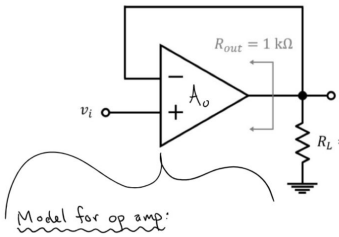
(c) $R_i = R_s/10$ $R_L = R_o/10$

$$\frac{v_o}{v_s} = \left(100\frac{V}{V}\right)\left(\frac{1R_s}{11R_s}\right)\left(\frac{1R_o}{11R_o}\right) = 0.8264\ V/V = -1.656\ dB$$

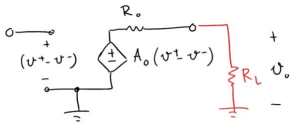
You're tasked with building the op amp circuit in figure PS3.1 and told the final circuit must have a maximum gain error of 0.5% when driving a load resistance of 100 Ω. If the op amp's output impedance is 1 kΩ, what is the minimum open loop gain it must have?



6.



Model for op amp:



$$v_o = A_o(v^+ - v^-) \cdot \frac{R_L}{R_o + R_L}$$

(this factor is due to finite R_o)

let's call $k = \frac{R_L}{R_o + R_L}$

$$v_o = kA_o(v_i - v_o)$$

$$\frac{v_o}{v_i} = \frac{1}{1 + \frac{1}{kA_o}} = A_v$$

$$\left.\frac{v_o}{v_i}\right|_{A_o = \infty} = A_{v,ideal} = 1$$

$$\text{gain error} = \frac{A_{v,ideal} - A_v}{A_{v,ideal}} \times 100 = \frac{1 - \left(\frac{1}{1 + 1/kA_o}\right)}{1} \times 100$$

$$\mathcal{E} = \frac{1 + \frac{1}{kA_o} - 1}{1 + \frac{1}{kA_o}} \times 100 = \frac{100}{1 + kA_o}$$

$$\mathcal{E} = 0.5\% = \frac{100}{1 + kA_o} \qquad k = \frac{R_L}{R_L + R_o} = \frac{100}{1.1 \times 10^3}$$

$$k = 0.091$$

$$kA_o = 199$$

$$A_o = \frac{199}{0.091} \Rightarrow A_o = 2189\ V/V$$

Computer Science

Easy

- In an array A of size 2N, there are N+1 unique elements, and exactly one of these elements is repeated N times. Return the element repeated N times.

```
def repeatedNTimes(self, A):
    count = collections.Counter(A)
    for k in count:
        if count[k] > 1:
            return k
```

- An array A of size N is structured such that its elements (integers) will increase before they start to decrease (think of a mountain). Find the largest element in the array.
    - Approaches
        - Naive: search in linear order until you see an element at index n+1 < at index n. Return val at index n.
            - Why is this not optimal?
            - What does the problem statement give us that we can use to shorten our search?
        - Better/Best
            - Binary approach
            - Ask why this works better
- Students are asked to stand in non-decreasing order of heights for an annual photo.
- Return the minimum number of students not standing in the right positions. (This is the number of students that must move in order for all students to be standing in non-decreasing order of height.)

```
def heightChecker(self, height):
    temp = sorted(heights)
    count = 0
    for i in range(len(temp)):
        if temp[i] != heights[i]:
            count += 1
    return count
```

- Peak index in a mountain array (array values ascend and then descend at some index)

```
def peakIndexInMountainArray(self, A):
    for i in xrange(len(A)):
        if A[i] > A[i+1]:
            return i
```

- Given the root node of a binary search tree, return the sum of values of all nodes with value between L and R (inclusive). The binary search tree is guaranteed to have unique values.
- Bit shifting: given a 32 bit unsigned int n, rotate the int to the right by k digits

    - Give them this smaller 8 bit example: (n, k) = (0011 1010, 3) -> (0100 0111)
    - Edge case: Give them this example if they give solution 1
        - (n, k) = (0011 1010, 12) -> (1010 0011)
    - Solution 1: (n >> k) | (n << (32 - k))
    - Solution 2 (accounts for edge case): (n >> (k mod 32)) | (n << (32 - (k mod 32)))
        - Fun fact: this is an intro arista networks technical question, round 1, platform engineering position

```
def rangeSumBST(self, root, L, R):
    """
    :type root: TreeNode
    :type L: int
    :type R: int
    :rtype: int
    """
    mysum = 0
    if root == None:
        return 0
    elif L > root.val and R >= root.val:
        mysum += self.rangeSumBST(root.right, L, R)
    elif R < root.val and L <= root.val:
        mysum += self.rangeSumBST(root.left, L, R)
    else:
        mysum += self.rangeSumBST(root.left, L, R) + self.rangeSumBST(root.right, L, R) + root.val
    return mysum
```

Given an N-ary tree, find its maximum depth

```
class Solution(object):
    def maxDepth(self, root):
        """
        :type root: Node
        :rtype: int
        """
        if root == None:
            return 0
        elif root.children == []:
            return 1
        else:
            list1 = []
            for child in root.children:
                list1.append(1 + self.maxDepth(child))
            return max(list1)
```

Medium

- Binary tree inorder traversal

```java
class Solution {
    public List < Integer > inorderTraversal(TreeNode root) {
        List < Integer > res = new ArrayList < > ();
        helper(root, res);
        return res;
    }

    public void helper(TreeNode root, List < Integer > res) {
        if (root != null) {
            if (root.left != null) {
                helper(root.left, res);
            }
            res.add(root.val);
            if (root.right != null) {
                helper(root.right, res);
            }
        }
    }
}
```

Given a non-empty 2D array `grid` of 0's and 1's, an **island** is a group of `1`'s (representing land) connected 4-directionally (horizontal or vertical.) You may assume all four edges of the grid are surrounded by water.

Find the maximum area of an island in the given 2D array. (If there is no island, the maximum area is 0.)

Example 1:

```
[[0,0,1,0,0,0,0,1,0,0,0,0,0],
 [0,0,0,0,0,0,0,1,1,1,0,0,0],
 [0,1,1,0,1,0,0,0,0,0,0,0,0],
 [0,1,0,0,1,1,0,0,1,0,1,0,0],
 [0,1,0,0,1,1,0,0,1,1,1,0,0],
 [0,0,0,0,0,0,0,0,0,0,1,0,0],
 [0,0,0,0,0,0,0,1,1,1,0,0,0],
 [0,0,0,0,0,0,0,1,1,0,0,0,0]]
```

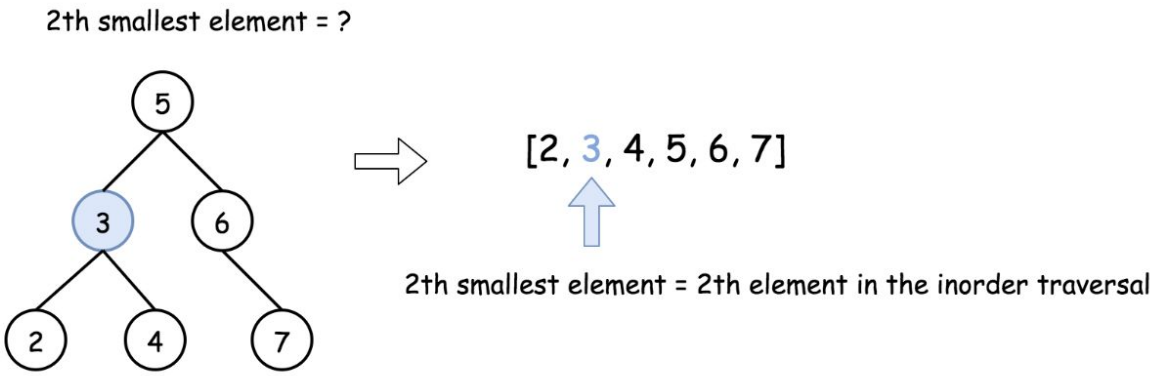
```
class Solution(object):
    def maxAreaOfIsland(self, grid):
        seen = set()
        def area(r, c):
            if not (0 <= r < len(grid) and 0 <= c < len(grid[0])
                    and (r, c) not in seen and grid[r][c]):
                return 0
            seen.add((r, c))
            return (1 + area(r+1, c) + area(r-1, c) +
                    area(r, c-1) + area(r, c+1))

        return max(area(r, c)
                   for r in range(len(grid))
                   for c in range(len(grid[0])))
```

- Finding the kth smallest unique integer in a tree

Approach 1: Recursion

It's a very straightforward approach with $\mathcal{O}(N)$ time complexity. The idea is to build an inorder traversal of BST which is an array sorted in the ascending order. Now the answer is the `k - 1` th element of this array.



Java | Python — Copy

```python
class Solution:
    def kthSmallest(self, root, k):
        """
        :type root: TreeNode
        :type k: int
        :rtype: int
        """
        def inorder(r):
            return inorder(r.left) + [r.val] + inorder(r.right) if r else []

        return inorder(root)[k - 1]
```

Hard

- Bit masking
  - Given a 32-bit integer, return the integer but with the pairwise bits swapped:
    - Give them an example: 00 10 01 11 01 10 -> 00 01 10 11 10 01
    - Solution: Iterative
      - Complex, hear them out but nudge them to the solution below
    - Solution: Bit masking
      - Create an odd mask: 0xaaaaaaaa
      - Create an even mask: 0x55555555
      - Shift odd mask to the left by 1, even to the right by 1

- Or them together; result is swapped bits
- Work with interviewee to make sure they get why this works
    - Fun fact: this is an arista networks technical question, round 2, platform engineering position

Behavioural Interview Questions

- How would you go about convincing your coworkers of an unpopular opinion?
- Did you ever fall short of your goals? Why?
- Describe a time when you had to work with someone you shared personal differences with.
- Describe a time when you successfully persuaded someone to see things your way
- Tell me about a time when you have been dissatisfied in school / work
- Describe a time you felt success while working in a group setting.
- Tell me about a time when you handled a challenging situation.
- Tell me about a time you set a goal for yourself. How did you go about ensuring that you would meet your objective?