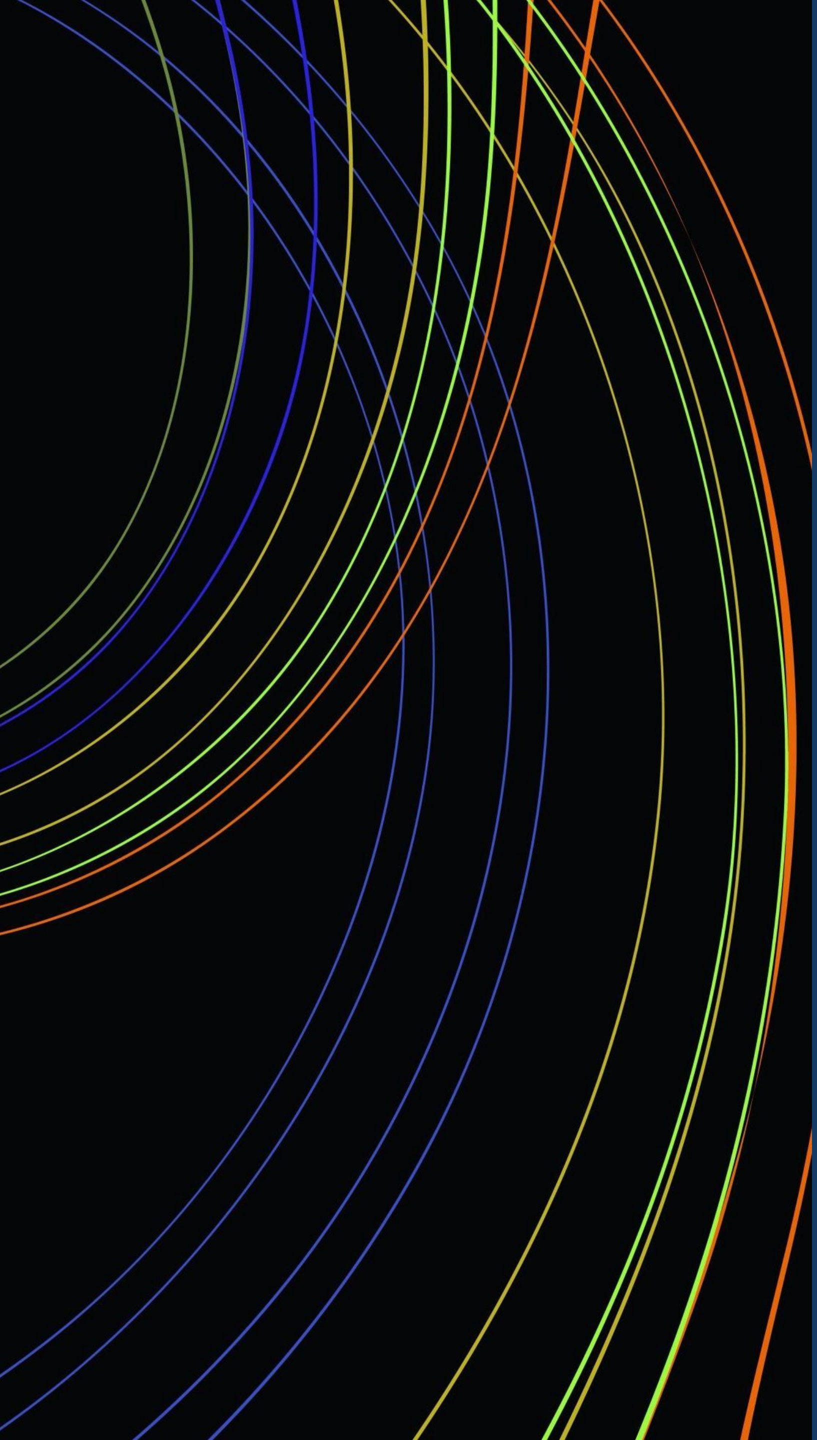
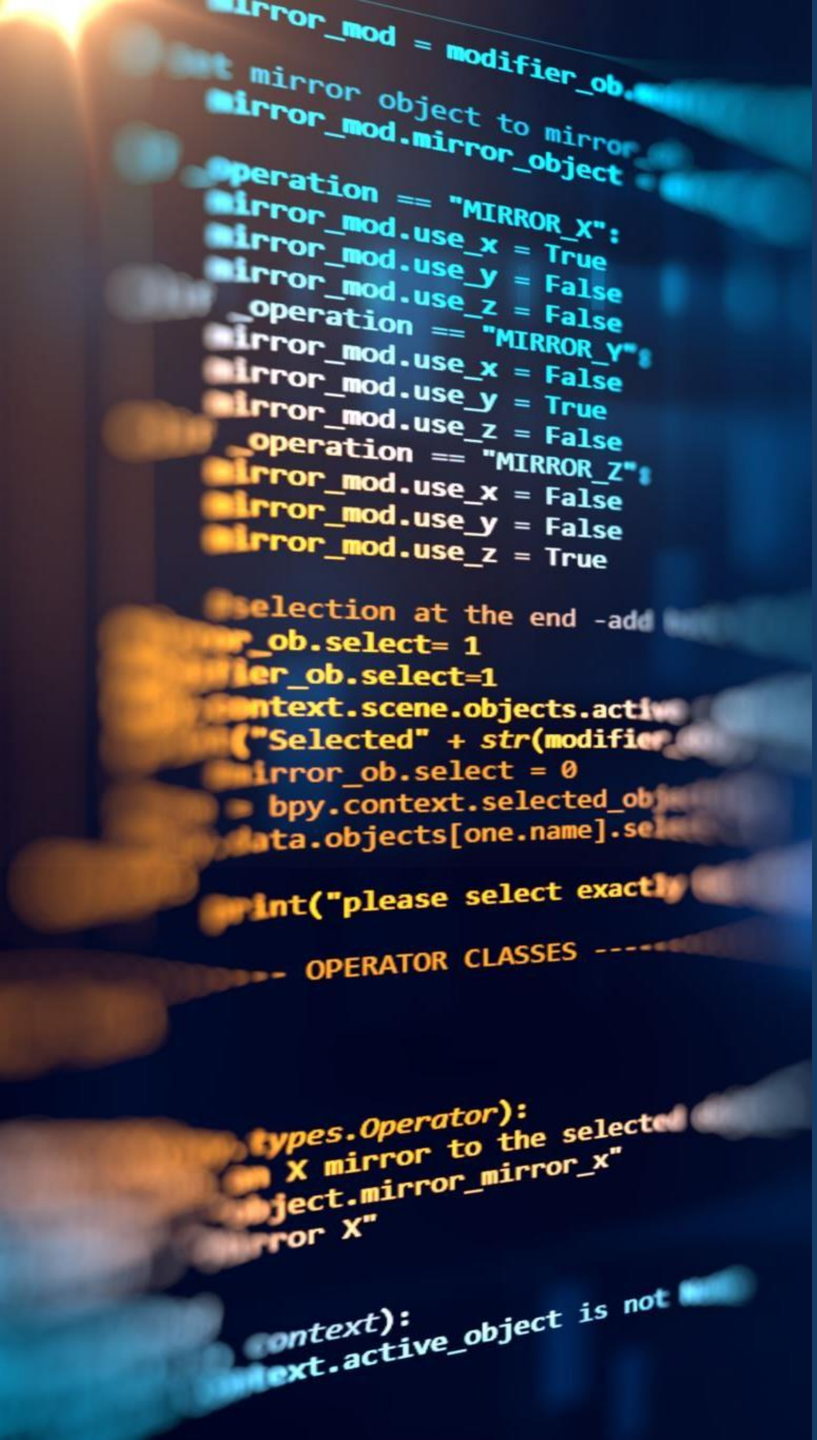


CRUSHING TECHNICAL INTERVIEWS

Anna Nyulund



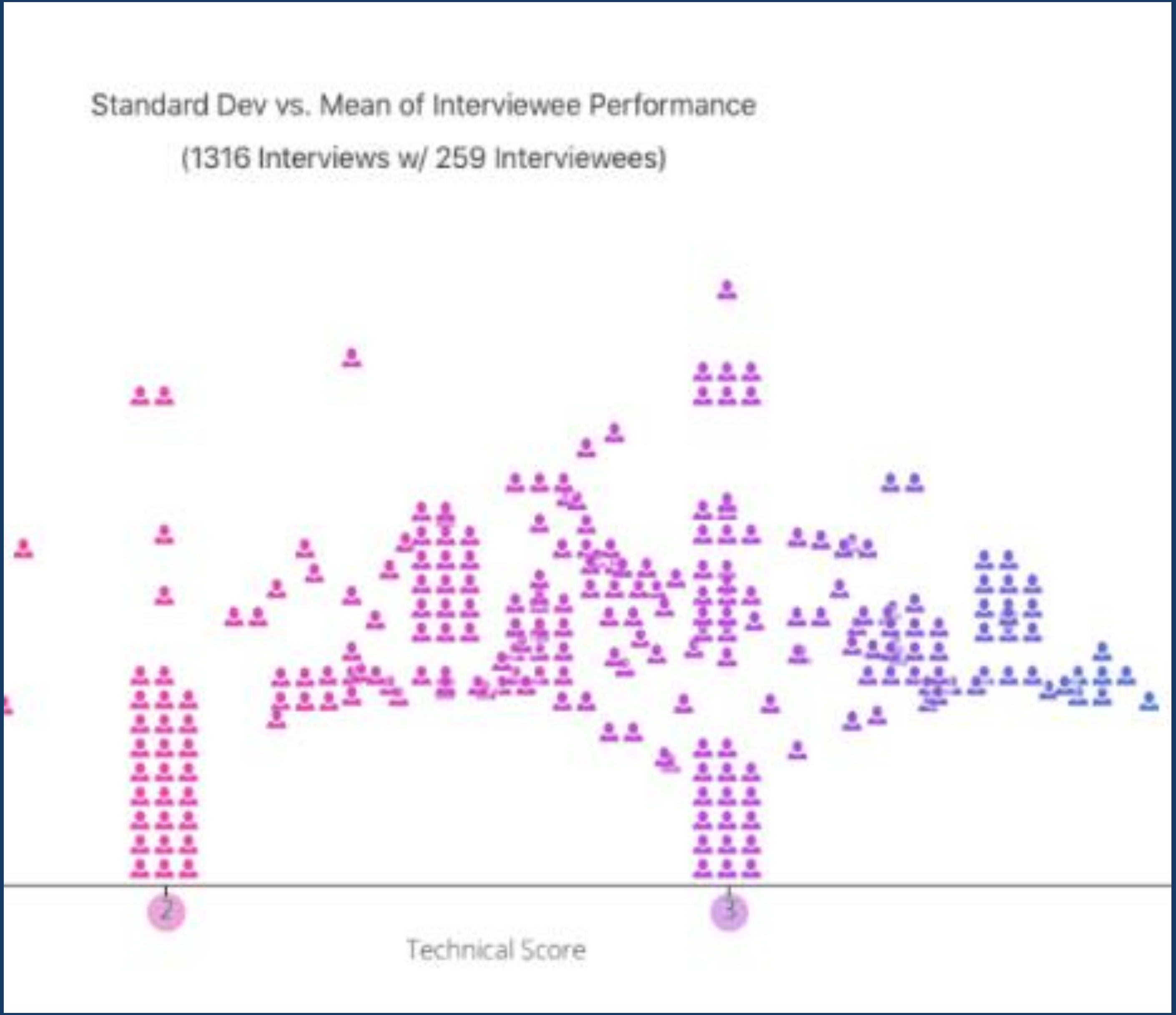
BACKGROUND

Common belief:

- Basic coding skills
- CS Algo and Data Structures knowledge
- Problem Solving smarts
- Software Engineering

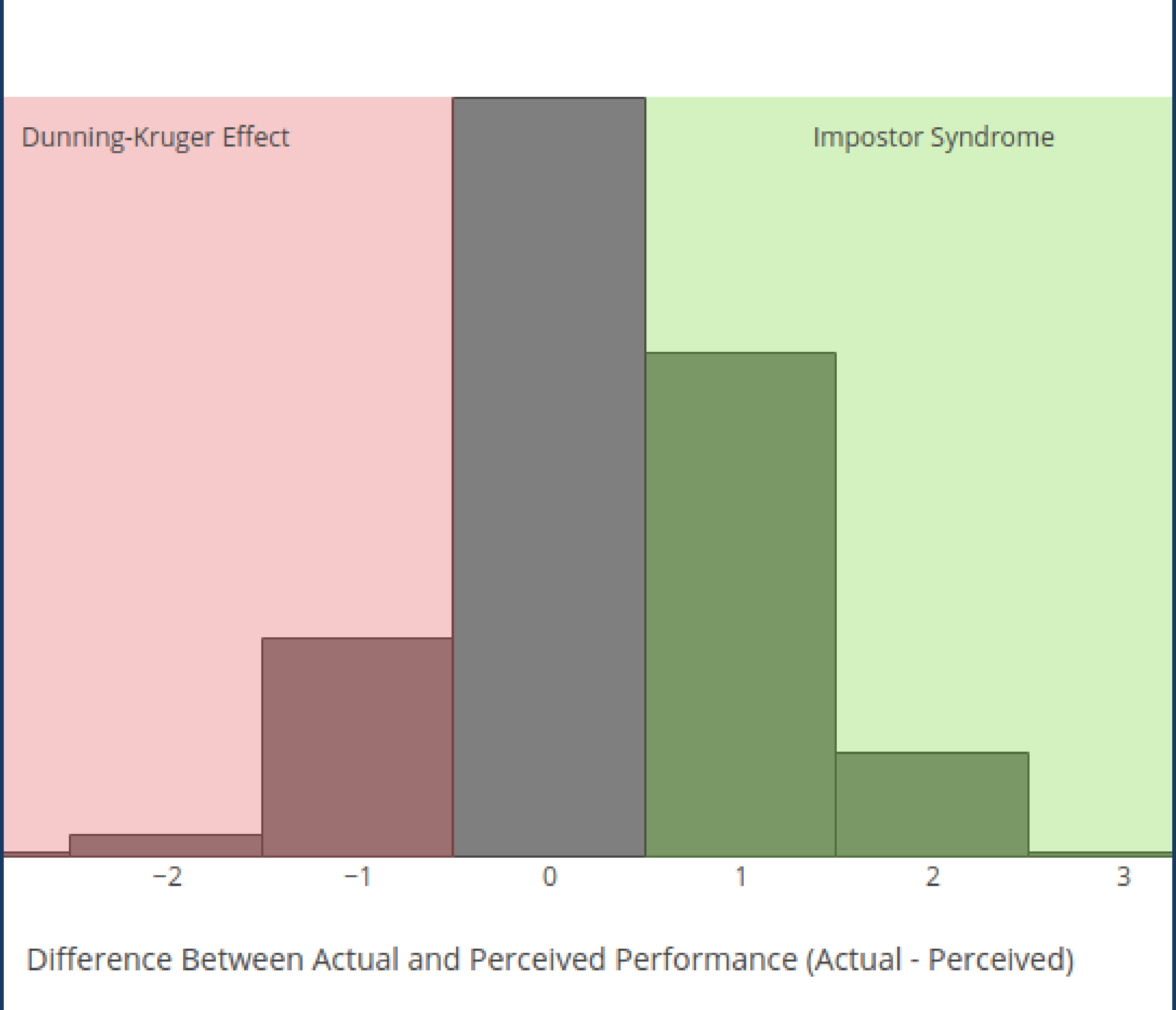
Uncommon belief:

- Whether a job candidate has performance anxiety rather than whether the candidate is competent at coding
- A process whose results are nondeterministic and often arbitrary
- Results are not repeatable and are not an indication of future performance



MEASURE OF PERFORMANCE

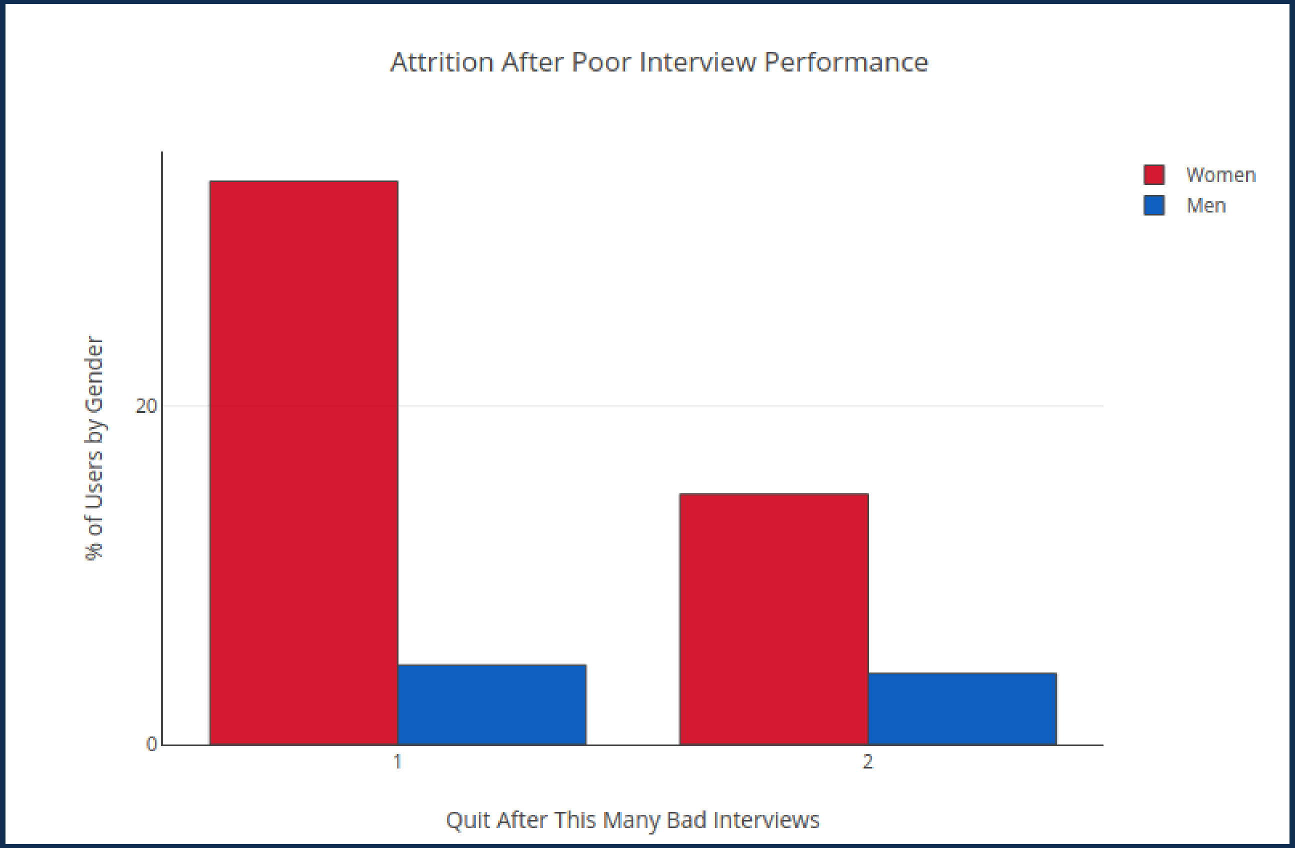
- Interview performance from interview varied, even for people with a high average performance. Roughly 25% are consistent.



GAUGING OWN PERFORMANCE

ISSUE OF DIVERSITY

- Women are 7 times more likely to stop practicing than men:



OVERCOMING STRESS AND ANXIETY

Anxiety causes stress. Stress decreases performance by approximately 50%

Rational, logical thinking goes out of the window. Cortisol increases, and heart rate increases. Tunnel vision prevails. Thoughts are racing, panic arises, you "freeze" or start rambling and thinking becomes clouded. Can get a headache and desire to argue or run away. You don't realize that your thinking is cloudy, because the thinking is cloudy.

Establish prospective pre-mortem hindsight to minimize the "disaster". Predict what could go wrong and prevent it.

Train hippocampus spatial memory that you can easily access during the interview for "lost things".

STRATEGY (PRE-MORTEM)

- Postpone interview until ready
- Create Mental Map of Problem Solving Strategies (like a bookshelf in your head)
- Ask for accommodations (double time, take home test, your own IDE, etc.)
- Coding Frequency
- Focus on main goal: show your flow
- Don't rush
- Break it down
- 80% "Muscle memory" and 20% modification

WHAT NOT TO DO

- Not accepting the situation
- Arguing with recruiters/interviewers
- Wasting time in self-pity
- Thinking that you are not good/real software engineer/ data scientist
- Thinking that you do not know anything
- Thinking that you cannot code
- Thinking that you are bad problem solver
- Become hopeless
- Give up (remember attrition rate in females in previous slides)

FACTS

More than 50% of Senior Engineers refuse the coding interviews or negotiate a waver

Many job candidates cheat if they can find a way

Many Engineering Managers/ Interviewers hate coding interviews as much as you do and view them as unnecessary obstacle

You will have more interviews some of which you fail and some of which you will pass for the rest of your career

PROBLEM-SOLVING MINDSET

- **Mental discipline:** ability to go on long, wild-goose chases and have open minded experimentation
  - **Timescale of concentration** required needs to increase
  - **Dealing with Failure:** investigation is always worthwhile even if it leads to dead-end
  - **Emotional attitude:** Mountaineer vs Gym Rat . "The explorer is the person who is lost"
  - **Problem-solving is a craft that can be learned or transferred**
    - Build background knowledge
    - Learn how others solved it
    - Active engagement

MENTAL MAP

Big O

Sorting Fundamentals

Hash tables (dictionaries in Python)

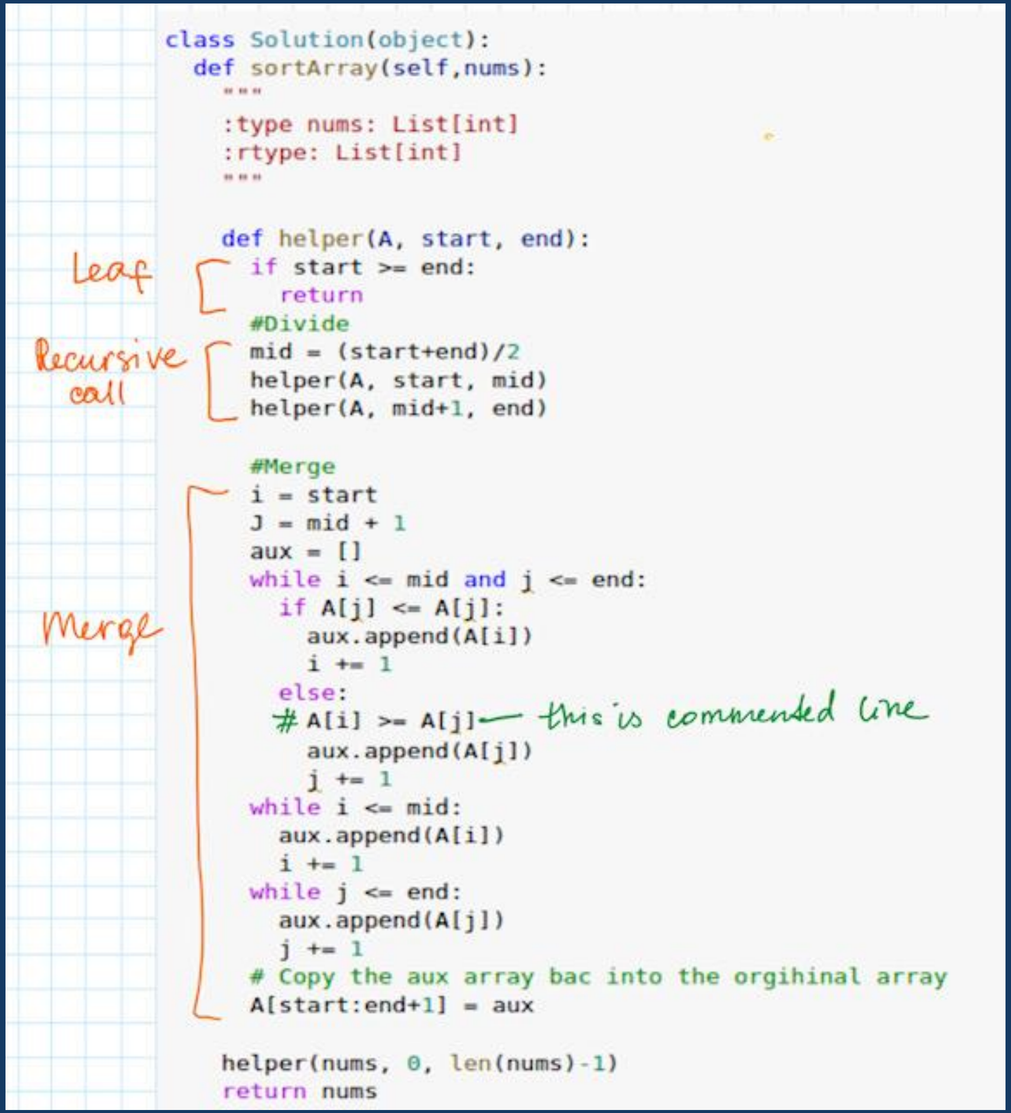
Trees

Graphs

NP-compete

DSA Annotated Reference

MEMORIZATION



```
class Solution(object):
  def sortArray(self,nums):
    """
    :type nums: List[int]
    :rtype: List[int]
    """

    def helper(A, start, end):
      if start >= end:
        return
      #Divide
      mid = (start+end)/2
      helper(A, start, mid)
      helper(A, mid+1, end)

      #Merge
      i = start
      J = mid + 1
      aux = []
      while i <= mid and j <= end:
        if A[j] <= A[j]:
          aux.append(A[i])
          i += 1
        else:
        # A[i] >= A[j]
          aux.append(A[j])
          j += 1
      while i <= mid:
        aux.append(A[i])
        i += 1
      while j <= end:
        aux.append(A[j])
        j += 1
      # Copy the aux array bac into the orgihinal array
      A[start:end+1] = aux

    helper(nums, 0, len(nums)-1)
    return nums
```

GENERAL ADVICE

Spend at least 5 min understanding the problem

Offer brute-force solution as quickly as possible

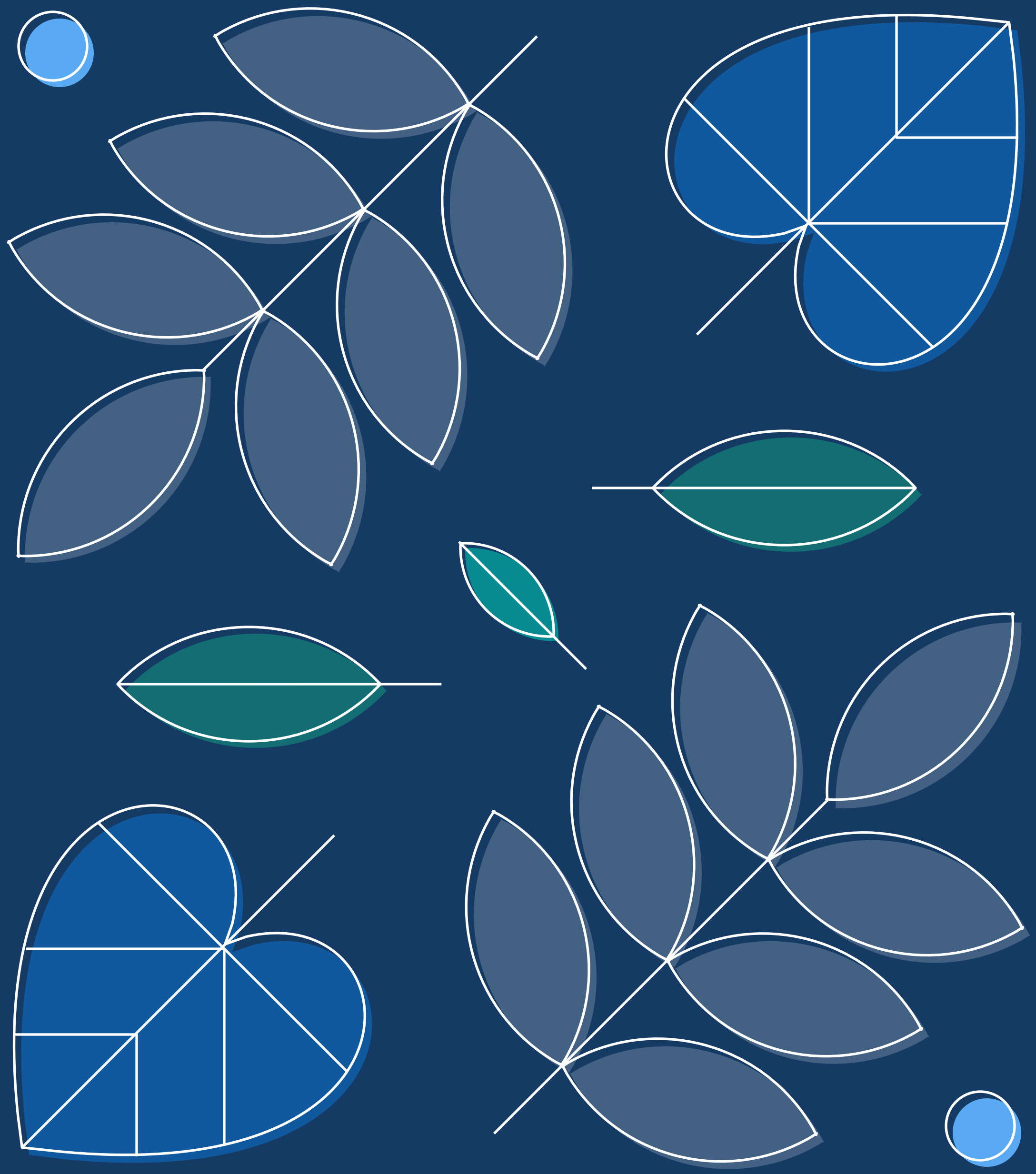
Don't worry about syntax errors, incorrect library usage unless interviewer prompts

Think about time and space complexities from onset, not as afterthought

- Frame conversation in trade-offs
- Light pseudocode before coding (Never jump straight to coding)
- Overcommunicate (need to show your flow and train of thought)
- Ask if they want compilabale code vs logic

STUDY RESOURCES

- Interview Kickstart Bootcamp
  - 3 months long
  - 6 months worth of materials accessible for a lifetime
  - Cover Data Structures, Leadership and System Design
  - Mock interviews
  - Weekly workshops
  - Referrals to FAANG



ONLINE RESOURCES

- Leetcode
    - Leethub
- Grokkin the Coding Interview
- Interview Cake
- Google Tech Dev Guide
- Cracking the coding interview
    - Resources
    - Big O presentation

ONLINE RESOURCES

- Article on Interview preparation
- High Scalability
- Google mock interview
- Facebook mock interview
- Top 10 algorithm interview questions

REFERENCES

- https://blog.interviewing.io/you-cant-fix-diversity-in-tech-without-fixing-the-technical-interview/
- https://www.engr.ncsu.edu/news/2020/11/11/tech-sector-job-interviews-assess-anxiety-not-software-skills-2/#:~:text=A%20new%20study%20from%20NC,candidate%20is%20competent%20at%20coding.