

Welcome to Data Structures & Algorithms

Or, how to pass technical interviews given by programmers

Terence Parr
MSDS program
University of San Francisco

From former cohort student

Hey Terence, I just had a technical interview with a mentorship program called . They asked me the Trie question you've mentioned so many times in class! In fact even the other question they asked was one of the problems you discussed - merging two sorted arrays

Course contents

<https://github.com/parrt/msds689>

- How to read code
- A formula for problem-solving simple algorithm problems
- Core data structures, a unifying perspective
- Algorithm complexity analysis
- “*So much recursion!*” – MSDS2019 student comment
- Walking and searching data structures
- Sorting (with all of my dirty tricks)
- Graphs and graph algorithms

Course projects

- Convert htable project to object-oriented version (8%)
 - With some extensions
 - ...and using somebody else's code from two years ago!
 - hint: it's kinda stinky code. ha!
- kmeans clustering, kmeans++ initial point selection (17%)
 - Spectral clustering using Breiman's unsupervised learning trick for RFs
 - Image compression applications
- Feature importance and selection (20%)
 - Permutation and drop column
 - Automatic feature selection
- Work as hard or as little as you want (I give no unit tests)
 - I will assign check, check-, check-- based upon your reports

<https://github.com/parr/msds689/blob/master/projects/oohtable/oohtable.md>

<https://github.com/parr/msds689/blob/master/projects/kmeans/kmeans.md>

<https://github.com/parr/msds689/blob/master/projects/featimp/featimp.md>

Student evaluation

*Please note grading will take at least a week
For projects, but I'll grade exams quickly*

Artifact	Grade Weight	Due date
OO hashtable	8%	Fri, Apr 2 10:00am
Clustering	17%	Sun, Apr 18 11:59pm
Feature selection and importance	20%	Sun, May 9 11:59pm
Exam 1	25%	3:30PM-4:30PM Fri Apr 16
Exam 2	30%	10AM-11:00AM Fri May 7

last day of class

Kmeans and feature importance projects

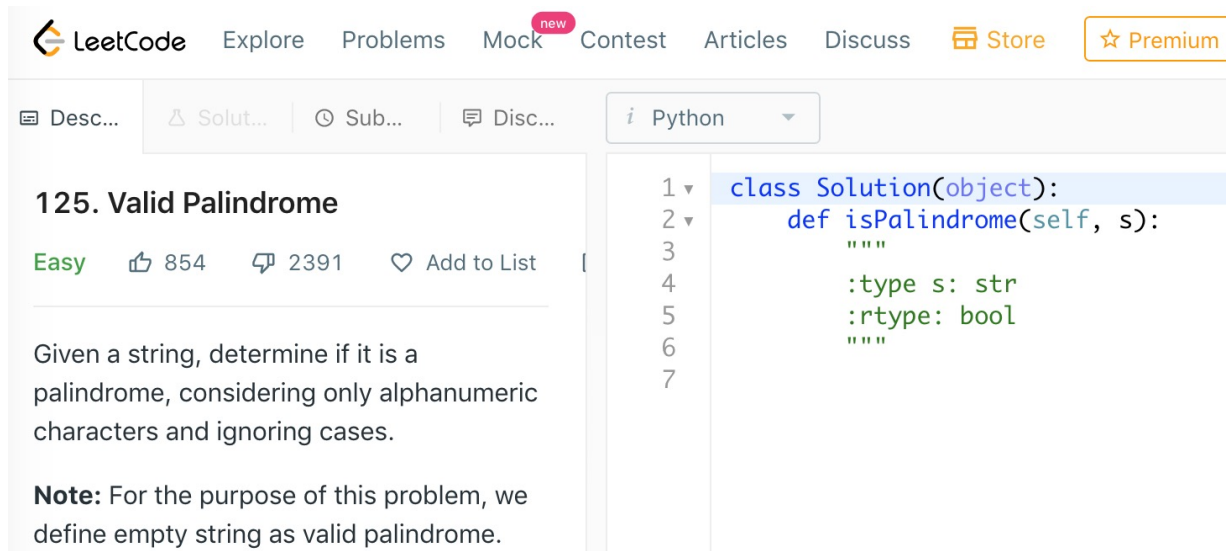
Practice writing and implementing algorithms

Hi Terence! Hope you're keeping safe! I wanted to let you know an interviewer wanted to see some reports/projects prior to our interview. I had shared our feature importance and kmeans project reports we worked on for our class. It was super helpful and they especially liked the feature importance one! Thank you for making us work on it and I think it will be good for future cohorts too 😊

to be added to your tracker, I got a question on kmeans in my phone interview.

Extra things you can do

- Lots of little practice quizzes; e.g.,
<https://github.com/parrt/msds689/blob/master/labs/quiz-oo.ipynb>
- LeetCode algorithm and data structures challenges. e.g.,
<https://leetcode.com/problems/valid-palindrome/>



The screenshot shows the LeetCode website interface. At the top, there's a navigation bar with links: LeetCode, Explore, Problems, Mock (with a 'new' badge), Contest, Articles, Discuss, Store, and Premium. Below this, there's a sub-navigation bar with tabs: Desc..., Solut..., Sub..., and Disc..., along with a language selector set to 'Python'. The main content area displays the problem '125. Valid Palindrome' with a difficulty level of 'Easy', 854 likes, 2391 discussions, and an 'Add to List' button. The problem description states: 'Given a string, determine if it is a palindrome, considering only alphanumeric characters and ignoring cases.' A note specifies: 'Note: For the purpose of this problem, we define empty string as valid palindrome.' On the right, a code editor shows a Python solution using a class:

```
1 class Solution(object):
2     def isPalindrome(self, s):
3         """
4         :type s: str
5         :rtype: bool
6         """
7
```

Resources

- A great free book on [algorithms by Jeff Erickson](#)
- Kleinberg and Tardos, *Algorithm Design*
 - Please see compressed pdf kleinberg-common-running-times.7z in Canvas course files area (do not post material publicly please)
- A very useful set of [programming-concepts-for-data-science](#) and [data science coding questions](#) by former USF MSDS student [Shikhar Gupta](#)
- [10 steps to solving a programming problem](#)
- Review [OO notebook](#) and [Operator overloading notebook](#)

Administrivia

- The usual academic honesty rules will be enforced; in projects, reports, exams or any other artifact; [Honor Code](#)
 - Do not represent another person's work as your own
 - Don't leave your laptop unattended/unlocked; others can take a picture of your code or simply use a USB key quickly
- Students with Disabilities
 - If you are a student with a disability or disabling condition, or if you think you may have a disability, please contact USF [Student Disability Services](#) (SDS) for information about accommodations.
 - If you are sick, please let us know beforehand, not after-the-fact
 - More details on the course syllabus: <https://github.com/parrt/msds689>

Why this course, why now?

- At least for the moment, many of the people interviewing you will be programmers, pretending to be data scientists
- What do they know? Programming, data structures, and algorithms
- Being able to organize data within a machine or across machines is a key skill for a data scientist
- The larger the data, the more critical it is to understand how to measure algorithm performance and how to design efficient solutions
- Optimally, you'd get this course much earlier, but the timing is good for your interviewing and was only spot we could jam this course in

How to get a job

1. Be accomplished, be interesting
 2. Know lots of people
 3. Mine social network looking for job
- When that fails or simultaneously
 - Cold apply to jobs via the web (a lot!)