Announcement: CTI One on LeetCode Now

What does LeetCode do

LeetCode is a platform for preparing technical coding interviews. Pick from an expanding library of more than 450 questions, code and submit your solution to see if you have solved it correctly. It is that easy!

Note: LeetCode platform currently supports a total of 11 languages: C, C++, Java, Python, C#, JavaScript, Ruby, Swift, Go, Bash, MySQL.

https://leetcode.com/

What does CTIOne do?

A boot camp for teaching and solving leetCode 600+ questions, to prepare readiness for technical coding interviews.

What is better: you don't submit your solution on line and wait for result, we teach you how to solve it and code it correctly. It is that easy!

Plus: we have embedded 100+ questions to prepare for technical interview on embedded systems and IoT.

Follow our AI Deep Learning Training Course on github:

https://github.com/hualili/opencv/tree/master/IP110-Deep-Learning

Announcement: CTI One on LeetCode Now (2)

What does LeetCode do

LeetCode is a platform for preparing technical coding interviews. Pick from an expanding library of more than 450 questions, code and submit your solution to see if you have solved it correctly. It is that easy! https://leetcode.com/

What does CTIOne do?

A boot camp for teaching and solving leetCode 600+ questions. For each question, we teach

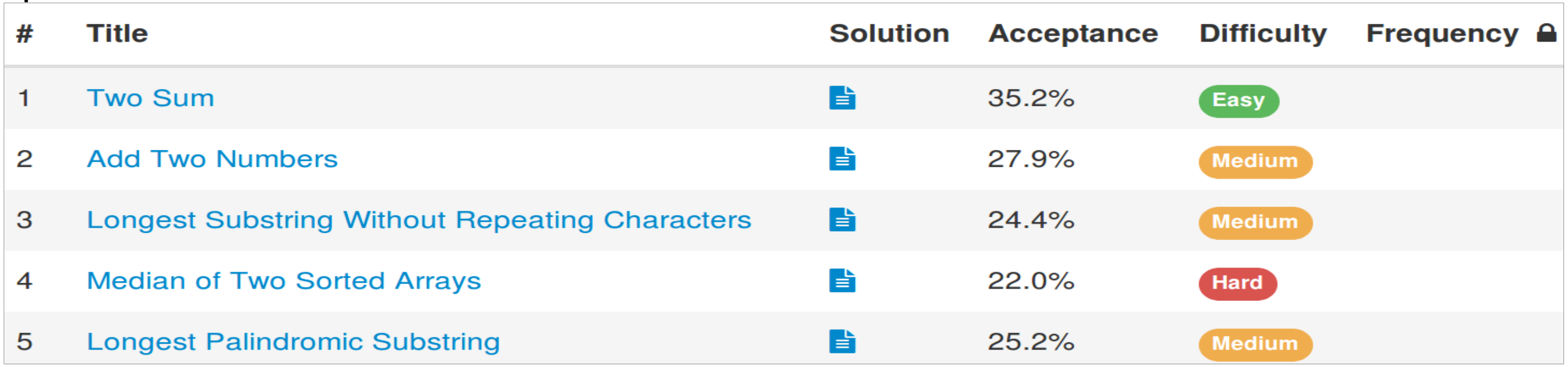
(1) the theory behind it;

(2) the coding tricks;

(3) which and what type of companies are likely to ask this questions.

You solve a question correctly, CTI One will offer you a free lunch gift card

Samples:



#	Title	Solution	Acceptance	Difficulty	Frequency 🔒
1	Two Sum	📄	35.2%	Easy	
2	Add Two Numbers	📄	27.9%	Medium	
3	Longest Substring Without Repeating Characters	📄	24.4%	Medium	
4	Median of Two Sorted Arrays	📄	22.0%	Hard	
5	Longest Palindromic Substring	📄	25.2%	Medium	

Preprocessing

Name of Module	Description	Execution and Application
1 augment_data.py	Augment cropped raw image data including Gaussian blur, motion blur, and Rotation to produce 20 new images.	$ python augment_data.py Note: raw data set directory path can be changed in program

```
import sys, os
import cv2

data_dir = "Training"  #path for the folder

#list all the directories which hold all different images in each directory

directories = [d for d in os.listdir(data_dir)
           if os.path.isdir(os.path.join(data_dir, d))]

for d in directories:
    images = []
    label_dir = os.path.join(data_dir, d)
    file_names = [os.path.join(label_dir, f)
              for f in os.listdir(label_dir)
              if f.endswith(".ppm") or f.endswith(".JPG")
              or f.endswith(".png")]

    # Augment the image dataset with rotation and blurring
```

Harry Li, Ph.D.

Preprocessing: Gaussian Blur

```
for f in file_names:
    img = cv2.imread(f)
    if img is not None:
        print("Processing" + f)
        M = cv2.getRotationMatrix2D((img.shape[1] / 2, img.shape[0] / 2),
                                    10, 1)  # rotation matrix by 10 degree
        rotate1 = cv2.warpAffine(img, M, (img.shape[1], img.shape[0]))
                                      # rotate image and assign it back
        M = cv2.getRotationMatrix2D((img.shape[1] / 2, img.shape[0] / 2),
                                    -10, 1) # rotation matrix counterwise
        rotate2 = cv2.warpAffine(img, M, (img.shape[1], img.shape[0]))
                                      # rotate image and assign it back

        blur1 = cv2.GaussianBlur(img, (5, 5), 3) # 5 by 5 kernel, sigma 3
        blur2 = cv2.GaussianBlur(img, (7, 7), 5)
        blur3 = cv2.GaussianBlur(img, (9, 9), 7)

        rotate1_blur1 = cv2.GaussianBlur(rotate1, (5, 5), 3)
        rotate1_blur2 = cv2.GaussianBlur(rotate1, (7, 7), 5)
        rotate1_blur3 = cv2.GaussianBlur(rotate1, (9, 9), 7)

        rotate2_blur1 = cv2.GaussianBlur(rotate2, (5, 5), 3)
        rotate2_blur2 = cv2.GaussianBlur(rotate2, (7, 7), 5)
        rotate2_blur3 = cv2.GaussianBlur(rotate2, (9, 9), 7)
        # Aug 10, 2017,add motion blur

        cv2.imwrite(f[0:-4] + "_rotate1.jpg", rotate1)
        cv2.imwrite(f[0:-4] + "_rotate2.jpg", rotate2)
        cv2.imwrite(f[0:-4] + "_blur1.jpg", blur1)
        cv2.imwrite(f[0:-4] + "_blur2.jpg", blur2)
```

1. 2D convolution
2. Kernel design
3. Gaussian kernel