challenges

November 20, 2021

1 Beginners' Hackathon - Python Challenges

Hi everyone! Thanks for joining the CUES Intro Hackathon 2021. Please find the questions and their marks below.

Unlike any other coding interviews, feel FREE to use Google, Reddit and other tools that help you solve the challenge. We are happy to see you pick up new python skills!

1.1 Section 1 [Easy level]

1.1.1 Q1: Taking the stairs to the Engineering Library (5pt)

In a sunny day in Michaelmas term, you just finished your lecture in LT0 and just wanna take a nap in the Engineering Library. You decided to take the staircase to the Engineering Library. There are `n` steps between the ground floor to the first floor. Hard as you may try, you could only take 1 step or 2 steps at a time. The question is, in how many distinct ways can you climb to the top?

Example: If `n == 3`, there are **three** ways to climb to the top. 1. 1 step + 1 step + 1 step 2. 1 step + 2 steps 3. 2 steps + 1 step

Hint: Write the first few numbers and see if you can find a pattern? Have you heard of the Fibonacci sequence?

Write a function to calculate the number of possible ways for `n` steps of staircase.

1.1.2 Q2: Giving away coupons (5pt)

After a social event, the CUES events team plans to give away coupons to the game winners. There are two winners: first prize and second prize. CUES prepared **even** number of coupons from different shops. We assign the coupons equally to them but want to maximise the **types** of coupon the first prize winner can get. Given the list of coupons with their shop ID, can you write a function to figure out how many coupons can the first prize winnder get?

Example 1:

Coupon list: `[1,1,1,2]`. There are four coupons, we can assign `[1,2]` to the first prize winner so they can get more types of coupons. So the programme returns `2`.

```
[ ]:
```

1.1.3 Q3: Pascal's triangle (5pt)

Given an integer `n`, return the first `n` Rows of Pascal's triangle.

In Pascal's triangle, each number is the sum of the two numbers directly above it as shown:

```
    1
   1 1
  1 2 1
 1 3 3 1
1 4 6 4 1
```

The output should be `[[1],[1,1],[1,2,1],[1,3,3,1],[1,4,6,4,1]]` for `n = 5`.

1.1.4 Q4: CUES Cipher (5pt)

The caesar cipher substitutes one character with another. To determine if one word has been encripted, can you dertermine first if two words are isomorphic.

Two strings `s` and `t` are isomorphic if the characters in `s` can be replaced to get `t`.

Example:

```
Input: s = "egg", t = "add"
Output: true
```

```
Input: s = "paper", t = "title"
Output: true
```

```
Input: s = "foo", t = "bar"
Output: false
```

1.1.5 Q5: Sort a list (10pt for 3 ways with intuitive explanation)

Can you find more than three ways to sort a list in Python? (You will have to explain how they work to our marker and pick the one you think is the most efficient.)

1.1.6 Q6: What's your number? (5pt)

CUES sponsorship officer is making calls to land new sponsors for this year. Can you varify a given string `s` if it is a vaild phone number?

You may assume that a valid phone number must appear in one of the following two formats: (xxx) xxx-xxxx or xxx-xxx-xxxx. (x means a digit). Any space in between is valid, for example (999) 999-9999 and "(99 9)99 9-9 9 99" are both valid.

Hint: `re` is a quite useful library!

1.1.7 Q7: Transpose (5pt)

Can you write a function to give transpose of an 2D array? (We extend the idea of transpose to any `m x n` 2D arrays)

Notes: It is also fine to use libraries!

Example:

```
Input: [[1,2,3],[4,5,6],[7,8,9]]
Output: [[1,4,7],[2,5,8],[3,6,9]]

Input: [[1,2,3,4], [5,6,7,8]]
Output: [[1,5],[2,6],[3,7],[4,8]]
```

1.1.8 Q8: Zen of Python (5pt)

Explore the mysterious library `this` and `antigravity`.

Use your own computer with Python 3 installed. Try to `import` these two modules.

1.2 Section 2 [Medium level]

1.2.1 Q1: Prime Numbers (10pt)

Given a positive integer N (N > 1), can you write a function to provide the number of prime numbers under N?

Bonus: can you provide a solution if N is very large?

1.2.2 Q2: How many days until birthday (10pt)

Borat entrusts you to design a programme to check how many days are there before his birthday. Can you help? The input date will be in the form of "DD/MM" (you don't have to validate the input). You can assume today is exactly 21 November.

Hint: I heard `datetime` was a very useful package.

1.2.3 Q3: Spiral, spiral, spiral (10pt)

Can you write a function to print out a spiral square `n x n` matrix?

```
Example: n = 3

1 2 3
8 9 4
7 6 5

Example: n = 4

01 02 03 04
12 13 14 05
11 16 15 06
10 09 08 07

(You also have to print out 0s to fill the position)
```

1.2.4 Q4: Cambridge Emails (10pt)

CUES is taking registrations for the next online event. Not all the registrants used their cambridge email accounts. Now that we have a list of emails addresses stored in the csv file (`emails.csv`). Can you write a function to help separate those non-crsid email accounts.

Note: the dataset was made up. P.S: You may want to use `pandas`

1.2.5 Q5: Password generator (10pt)

Use `random` library to create a function which will return a random password that meets the following conditions: - Password length must be 8. - Password must contain at least 1 upper case letters, 1 digit, and 1 special symbol. - Position of upper case, digit and special characters must not be fixed

1.2.6 Q6: Word Cloud (15pt)

Can you generate a word cloud consisting of country names by their rankings in the medal table in the Tokyo Olympic games? Be creative!

Hint: You may find the `wordcloud` library very helpful! The wordcloud takes a text and analyse the frequency of the words, can you generate a text consist of the country name with the frequency being their total medal numbers?

Link: https://olympics.com/en/olympic-games/tokyo-2020/medals

Example of a word cloud:

1.2.7 Q7 & Q8: Rona is not welcomed (20 points)

CUES wanted to know the pandemic trend in Cambridge so that the events team can plan accordingly. Can you use the API provided by the UK govenment and use the matplotlib to provide a visualisation of new cases in Cambridge?

Useful webpage: https://coronavirus.data.gov.uk/details/developers-guide/main-api

Python sample code provided:

[ ]:
```
from requests import get
from json import dumps


ENDPOINT = "https://api.coronavirus.data.gov.uk/v1/data"
AREA_TYPE = "nation"
AREA_NAME = "england"

filters = [
    f"areaType={ AREA_TYPE }",
    f"areaName={ AREA_NAME }"
]

structure = {
    "date": "date",
    "name": "areaName",
    "code": "areaCode",
```

```
    "dailyCases": "newCasesByPublishDate",
    "cumulativeCases": "cumCasesByPublishDate",
    "dailyDeaths": "newDeaths28DaysByPublishDate",
    "cumulativeDeaths": "cumDeaths28DaysByPublishDate"
}

api_params = {
    "filters": str.join(";", filters),
    "structure": dumps(structure, separators=(",", ":"))
}

response = get(ENDPOINT, params=api_params, timeout=10)

if response.status_code >= 400:
    raise RuntimeError(f'Request failed: { response.text }')

print(response.url)
print(response.json())
```

1.2.8 Q9: UN Security Council Representation (15pt)

The following website details the years in which each country has held a seat on the UN Security Council: https://www.un.org/securitycouncil/content/countries-elected-members

This Wikipedia page lists the populations of each UN member: https://en.wikipedia.org/wiki/List_of_countries_by_population_(United_Nations)

Accepting the above links as the only inputs, output the composition of the Security Council for every year since 1946, as well as the ratio of each member's individual population to the total population represented by the Security Council that year (as a percentage). Using the given 2019 population figures from the Wikipedia page is fine (i.e. ignore population changes). Don't forget the Permanent Members!

1.3 Section 3 [Applications]

1.3.1 Sudoku game PlayStation

In classic sudoku, the objective is to fill a **9×9** grid with digits so that **each column, each row, and each of the nine 3×3 subgrids** that compose the grid (also called "boxes", "blocks", or "regions") contains **all** of the digits from 1 to 9. The puzzle setter provides a partially completed grid, which for a well-posed puzzle has a single solution.

In this challenge, you will build a python sudoku game platform.

[ ]:
```
sudoku_set = [[5,3,_,_,7,_,_,_,_],
              [6,_,_,1,9,5,_,_,_],
              [_,9,8,_,_,_,_,6,_],
              [8,_,_,_,6,_,_,_,3],
              [4,_,_,8,_,3,_,_,1],
              [7,_,_,_,2,_,_,_,6],
```

```
            [_,6,_,_,_,_,_,2,8,_],
            [_,_,_,_,4,1,9,_,_,5],
            [_,_,_,_,_,8,_,_,_,7,9]]
```

Q1: Validate input for a position (10pt) Can you complete the method `check_validity` that takes the position `x`, `y` and an input `num` and verifies if `num` can be put in the location `x` and `y`.

Q2: Provide hints (10pt) Can you complete the method `get_hint` to provide possible output for one position `x`, `y`? Make use of the function you have written.

Q3: Update game status (10pt) Can you complete the method `update_game_status` to see if the game has finished or in a dead end (i.e., none of the empty positions has available candidates).

Q4: Play the game! (15pt) Can you complete method `play_game` to put together the game logic and keep the game running until the game ends? The game should be able to take inputs of x, y and num from the user and provides appropriate feedback of the given inputs.

[ ]:
```
class sudoku():

  def __init__(self, problem_set):
    self.board_ = problem_set
    self.steps_ = 0
    # 0 for unfinished
    # 1 for winning
    # 2 for losing
    self.status_ = 0
    self.word_bank_ = ["has not finished", "over, you won!", "over."]

  def __str__(self):
    res = f"Round {self.steps_}: Game {self.word_bank_[self.status_]}\n"
    res += "                y           \n"
    res += "     0 1 2   3 4 5   6 7 8  \n"
    res += "    ======================= \n"

    for i in range(9):
      if i == 4:
        res += f"x {i}|"
      else:
        res += f"  {i}|"
      for index, num in enumerate(self.board_[i]):
        res += f" {num}" if num else " _"
        if index % 3 == 2:
          res += " |"
      res += '\n'
      if i % 3 == 2:
        res += "    ----------------------- \n"
```

```
    return res

  def check_game_finished(self):
    # Write a function to check if the game is finished.
    # There should be three status of the game:
    # In progress
    # Won (all blanks finished without breaking the rule)
    # Failed (Rules have been broken for some positions)

    return True

  def update_game_status(self):
    # Check if the game has finished
    # If finished, then wining, because each step is valid

    # Check if unfinished, if there is no available stable state there
    # Otherwise, it is unfinished


  def check_validity(self, x, y, num):
    # Write a function to check if the state is valid for a sudoku
    # Check if input is legal

    # Check if the position is empty

    # Check rows

    # Check column

    # Check 3x3 grid

    return True

  def set_state(self, x, y, num):
    # Write a function to set a position to a given number

  def get_hint(self, x, y):
    # Write a function to provide a list of candidates for one position

    return ret

  def solve(self):
    # Write a function to solve the game with the above helper
    pass

  def play_game(self):
```

```
    # Write a procesure that you can play the game until the game is over
    pass
```

[ ]:
```
game = sudoku(sudoku_set)
print(game)
```

```
Round 0: Game has not finished
              y
     0 1 2   3 4 5   6 7 8
    ======================
  0| 5 3 _ | _ 7 _ | _ _ _ |
  1| 6 _ _ | 1 9 5 | _ _ _ |
  2| _ 9 8 | _ _ _ | _ 6 _ |
    ----------------------
  3| 8 _ _ | _ 6 _ | _ _ 3 |
x 4| 4 _ _ | 8 _ 3 | _ _ 1 |
  5| 7 _ _ | _ 2 _ | _ _ 6 |
    ----------------------
  6| _ 6 _ | _ _ _ | 2 8 _ |
  7| _ _ _ | 4 1 9 | _ _ 5 |
  8| _ _ _ | _ 8 _ | _ 7 9 |
    ----------------------
```

1.3.2 SHIELD Agent 101

The head quarter of S.H.I.E.L.D just recieved an encrypted message from black widow about the new criminal. Bit of a short time, widow didn't get the chance to encrypt the message via an advanced method. It seems all the characters have been swapped. There must be someway to interpret it. Can you help decode the message?

Message:

```
rxcm gptrykt zvktr fygxctrkpryf acataps pevsvrvyt fakvcl iuvsfuxxf. pzryk jpriuvcl p ixjexm tu

uy ruyc eylpc p bkxlkpg xz xetykqvcl ruy zvlurvcl ryiucvdayt xz cagykxat ixtragyf uykxyt pcf q
```

Letter frequency analysis (10pt) In English, with a large sample space the letter frequency always gives clues about the mapping between the new letter and the original one. Can you write a function to calculate the frequency of each letter from widow's message and fill in the table below? Let's see if it can give us some insights into the text.

Letter	a	b	c	d	e	f	g	h	i	j	k	l	m
Frequency(%)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Letter	n	o	p	q	r	s	t	u	v	w	x	y	z
Frequency(%)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

[ ]:
```
def letter_frequency_analysis(text):

    # Put your code here
    # Put your code here

    return None
```

Reference Table (10pt) Great! If only we could have a frequency table for normal English texts (`reference_texts.txt`). Now that you have this function done, can you get some long paragraph from the internet and make a table for the normal texts?

P.S: remember to convert it into lower cases before your analysis!

[ ]:
```
def read_file(path):

    # Put your code here
    # Put your code here

    return ""
```

Good clues always lead to another clue (20pt) Compare the two tables, can you give a wild guess for **5 mapping pairs**? Widow always says, Good clues always lead to another clue. Of course, we need to substitute back to see if they make sense. Perhaps we will discover more after that.

Just one more thing, to avoid confusion between the encoded letters and decoded letters, we'd better use the **capital** letter for the former and **lower** letter for the latter.We can create a `dictionary` to store these mappings, what do you say?

Hint: Create a new string everytime you decode in case something went wrong.

[ ]:
```
code_table = {'A': None, 'B': None, 'C': None, 'D': None, 'E':None, 'F':None,␣
↪'G':None,
              'H': None, 'I': None, 'J': None, 'K': None, 'L':None, 'M':None,␣
↪'N':None,
              'O': None, 'P': None, 'Q': None, 'R': None, 'S':None, 'T':None,␣
↪'U':None,
              'V': None, 'W': None, 'X': None, 'Y': None, 'Z':None}

def translate(text, code_table):

    # Convert the text to captical first
    text = text.upper()
```

```
    # Put your code here
    # Put your code here
    # Put your code here

    return text
```

Nailed it (20pt) Try a few guesses until you get 8 of the letters guessed. Here are some tips you may consider:

- Consider the next most frequent letter in the statistics vs. that in the encrypted message.
- In most cases, 1-letter word should be either `a` or `i`.
- In most cases, 1-letter after the apostrophe should be either `s` or `d`.
- Partially guessed short words of 2 or 3 letters gives a great hint of what the remaining letters are.
- Widow was using American spelling: "…zation" instead of "…sation".

Repeat the process untill you complete the decoding or you cannot continue. Can you put down your discoveries in the table below.

What is the confidential information about?

Letter	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Stubstitute																										

Widow, you can do better (Bonus 10pt) Letter E is the most occurring letter in English text. One of our agents proposes to substitute letter E by two different letter randomly, so that the effectiveness of letter frequency analysis is reduced. So for example, if the substitution for letter a to e is:

Letter	a	b	c	d	e
Substitute	x	a	t	m	r/k

The word **decade** can then be encrypted as **mrtxmr**, **mrtxmk**, **mktxmr** or **mktxmk**. However since two letter is used to encrypt one letter, we can only encrypt 25 different letters.

New agent, can you suggest one way to resolve this so that we can encrypt all 26 English letters. Illustrate your idea by generating an arbitrary substitution table for letter a to z.

Letter	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Stubstitute																										