How is it going?

https://forms.gle/iTkhayWMiyp4BRiR8

SCAN ME

Week 5

Anagrams and Missing Anagrams

Outline

- Assignment 1 – Point 3
  - Identifying Anagrams

- Assignment 1 – Point 4
  - Identifying Missing Anagrams

Identifying Anagrams

- To identify sentences that are anagrams of one another you need to keep track of the number of each character in each sentence.
- A possible solution is to create a 2D array of integers of size n x m.
  - n should be the number of sentences (or lines) in the input file
  - m should be equal to 26, i.e. the number of characters in the alphabet

For Example (1/3)

If the 2D array where you stored your sentences looks like the following

A	c	t	\0							
c	u	d	d	l	e	\0				
H	E	y		t	h	e	r	e	!	\0

The array counting the characters will look like the following:

1	0	1	0	0	0	0	0	0	0	0	0	0	. . .
0	0	1	2	1	0	0	0	0	0	0	1	0	. . .
0	0	0	0	3	0	0	2	0	0	0	0	0	. . .
a	b	c	d	e	f	g	h	i	j	k	l	m	

For Example (2/3)

If the 2D array where you stored your sentences looks like the following

A	c	t	\0							
c	u	d	d	l	e	\0				
H	E	y		t	h	e	r	e	!	\0

Element in position [1,3] indicates the number of 'd's in "cuddle"

1	0	1	0	0	0	0	0	0	0	0	0	0	. . .
0	0	1	2	1	0	0	0	0	0	0	1	0	. . .
0	0	0	0	3	0	0	2	0	0	0	0	0	. . .
a	b	c	d	e	f	g	h	i	j	k	l	m	

For Example (3/3)

If the 2D array where you stored your sentences looks like the following

A	c	t	\0							
c	u	d	d	l	e	\0				
H	E	y		t	h	e	r	e	!	\0

Element in position [2,4] indicates the number of ‘e’s in “Hey there!”

1	0	1	0	0	0	0	0	0	0	0	0	0	. . .
0	0	1	2	1	0	0	0	0	0	0	1	0	. . .
0	0	0	0	$\boxed{3}$	0	0	2	0	0	0	0	0	. . .
a	b	c	d	e	f	g	h	i	j	k	l	m	

A Few Tips on Identifying Anagrams

- After you create a data structure to save the number of characters of each sentence, you will have to compare sentences with one another and verify whether they have the same number of characters.

- To store information about the anagrams that you identified you can create a 2D array of integers, where each row contains the indexes of the sentences that are anagrams with one another.

For Example …

- If this is the sorted list of sentences:

  Act
  cat
  Computer science
  cuddle
  duck
  Hey there!
  I am Lord Voldemort
  Leonardo da Vinci! The Mona Lisa!
  O, Draconian devil! Oh, lame saint!
  Old Immortal dovers
  Software engineering
  tac
  Tom Marvolo Riddle
  UCD

For Example …

- If this is the sorted list of sentences:

Act
cat
Computer science
cuddle
duck
Hey there!
I am Lord Voldemort
Leonardo da Vinci! The Mona Lisa!
O, Draconian devil! Oh, lame saint!
Old Immortal dovers
Software engineering
tac
Tom Marvolo Riddle
UCD

- Your Array Storing Anagrams Should Look like the Following

0	1	11	\0
6	12	\0	
7	8	\0	

The null character can indicate that there are no more anagrams in the list

For Example …

- If this is the sorted list of sentences:

Act
cat
Computer science
cuddle
duck
Hey there!
I am Lord Voldemort
Leonardo da Vinci! The Mona Lisa!
O, Draconian devil! Oh, lame saint!
Old Immortal dovers
Software engineering
tac
Tom Marvolo Riddle
UCD

- Your Array Storing Anagrams Should Look like the Following

0	1	11	\0
6	12	\0	
7	8	\0	

It may be handy to keep track of a counter indicating how many groups of anagrams you identified

3 groups of anagrams identified

Another Trick

- To improve performance you can keep track in a separate array of the number of characters of each sentence.

- Note that you may have to ignore spaces, punctuation and other special characters.

- If 2 sentences have different lengths they cannot be anagrams of one another.

Keep Track of the Length of the Sentences

3	Act
3	cat
15	Computer science
6	cuddle
4	duck
8	Hey there!
16	I am Lord Voldemort
26	Leonardo da Vinci! The Mona Lisa!
26	O, Draconian devil! Oh, lame saint!
17	Old Immortal dovers
19	Software engineering
3	tac
16	Tom Marvolo Riddle
3	UCD

Only sentences that have the same length could potentially be anagrams of one another.

Identifying Missing Anagrams

A sentence (e.g., cuddle) is a missing anagram of another (e.g., UCD) if and only if:

- It is longer
- After removing a number of characters necessary to have the 2 sentences of the same length, they are anagrams of one another
  - For example, after transforming “cuddle” to “cud”, this is now an anagram of “UCD”

How to do it?

Identifying Missing Anagrams

Assume we have an array counting the number of characters of each sentence.

3	Act
3	cat
15	Computer science
6	cuddle
4	duck
8	Hey there!
16	I am Lord Voldemort
26	Leonardo da Vinci! The Mona Lisa!
26	O, Draconian devil! Oh, lame saint!
17	Old Immortal dovers
19	Software engineering
3	tac
16	Tom Marvolo Riddle
3	UCD

Identifying Missing Anagrams

cuddle is longer than UCD

3	Act
3	cat
15	Computer science
6	cuddle
4	duck
8	Hey there!
16	I am Lord Voldemort
26	Leonardo da Vinci! The Mona Lisa!
26	O, Draconian devil! Oh, lame saint!
17	Old Immortal dovers
19	Software engineering
3	tac
16	Tom Marvolo Riddle
3	UCD

Identifying Missing Anagrams

cuddle	0	0	1	2	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0
UCD	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
	a	b	c	d	e	f	g	h	j	k	i	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z

- Let’s remove the last 3 characters from cuddle
- Create a separate array counting the number of characters of cud

cud	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
UCD	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
	a	b	c	d	e	f	g	h	j	k	i	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z

So you can print the following “cuddle is a missing anagram of ucd if 3 characters removed”