HW10

HW10

HW10(Due on Dec. 10)

Find the most similar keyword!

- Implement the LCS algorithm for keywords
- Add each keyword into an array/linked list
- Given a string *s*, output the keyword *k*, such that *k*'s value and *s* have the longest common sequence among all the added keywords.

Requirements

- Maintain a keyword list, and implement the LCS algorithm
- For the list structure, you can
 - Use java.util.ArrayList
 - Or develop it by yourself

Operations

operations	description
add(Keyword k)	Insert a keyword k to an array
find(String s)	Find and output the most similar keyword by using the LCS algorithm

Keyword

A keyword is a tuple of [String name, Integer count]

```
    For example:
    {
        name: "Fang",
        count: 3
    }
```

- A keyword should output in format [name,count]:
 - [Fang,3]

I/O Example: add

- To do: Insert a keyword [k,c] to the list
- Input:
 - Token1: a constant "add"
 - Token2 : keyword name k
 - Token3 : keyword count c
 - EX: add Fang 3

I/O Example: find

- To do: Find and output the most similar keyword by using the LCS algorithm
- Input:
 - Token1: a constant "find"
 - Token2: a string s
 - EX: find NTU
- Output:
 - If list is empty, then output "InvalidOperation":
 - InvalidOperation
 - If it is legal:

NTU: [NCCU, 2]

Input file

- You need to read the sequence of operations from a txt file
- The format is firm
- Raise an exception if the input does not match the format

```
add Fang 3
add Yu 5
add NCCU 2
add UCSB 1
add Management 4
add Information 5
find NTU
find Manager
```

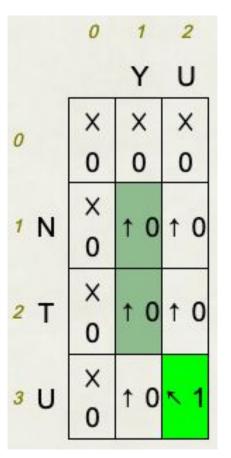
LCS

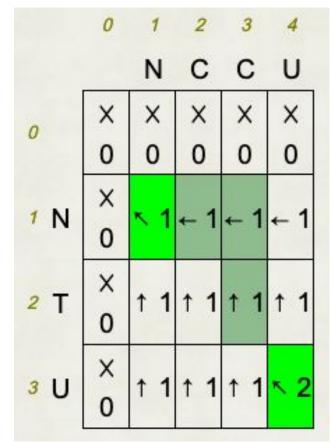
An LCS Algorithm

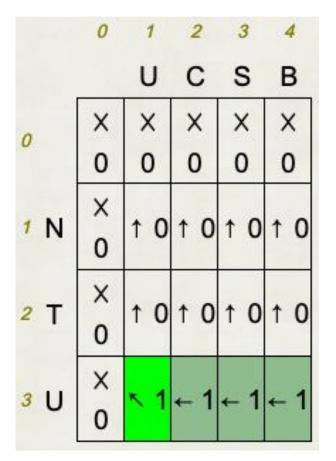
```
Algorithm LCS(X,Y):
Input: Strings X and Y with n and m elements, respectively
Output: For i = 0, ..., n-1, j = 0, ..., m-1, the length L[i, j] of a longest string
          that is a subsequence of both the string X[0..i] = x_0x_1x_2...x_i and
          the string Y [0.. j] = y_0y_1y_2...y_i
for i = 0 to n-1 do
     L[i,-1] = 0
for j = 0 to m-1 do
     L[-1,j] = 0
for i = 0 to n-1 do
     for j = 0 to m-1 do
          if x_i = y_i then
                L[i, j] = L[i-1, j-1] + 1
          else
                L[i, j] = \max\{L[i-1, j], L[i, j-1]\}
return array L
```

LCS

	0		2	3	4	
		F	Α	N	G	
0	×	X	X	×	X	
	0	0	0	0	0	
1 N	X 0	1 0	↑ 0	₹ 1	← 1	
2 T	X 0	† O	↑ 0	↑ 1	↑ 1	
<i>3</i> U	X 0	1 0	1 0	↑ 1	↑ 1	







LCS

	0	1	2	3	4	5	6	7	8	9	10		
		М	Α	N	Α	G	Е	М	Е	N	T		
0	X	×	×	X	X	×	×	×	X	×	X		
	0	0	0	0	0	0	0	0	0	0	0		
1 M	×	× 1	_ 1	← 1	← 1	← 1	← 1	× 1	← 1	← 1	← 1		
	0					- 1		` '		- 1			
2 A	×	↑ 1	₹ 2	← 2	× 2	← 2	← 2	← 2	← 2	← 2	← 2		
	0			•		_					_		
3 N X	×	↑ 1	† 2	× 3	← 3	× 3	← 3						
	0		1100										
4 A X	×	↑ 1	× 2	↑ 3	× 4	← 4	← 4	← 4	← 4	← 4	← 4		
	0	0 ' '						, ,					
5 G	5 G	×	↑ 1	↑ 1	1 2	↑ 3	1 4	₹ 5	← 5	← 5	← 5	← 5	← 5
	0		-		1. 7		•			, ,	. 0		
6 E	×	↑ 1	↑ 2	↑ 3	1 4	↑ 5	₹ 6	← 6	× 6	← 6	← 6		
	0												
7 R	×	↑ 1	↑ 2	↑ 3	1 4	† 5	1 6	↑ 6	↑ 6	1 6	↑ 6		
	0					, ,	, 0	, 0	, 3	, 0			

Output

```
NTU: [NCCU,2]
Manager: [Management,4]
```

Bonus HW

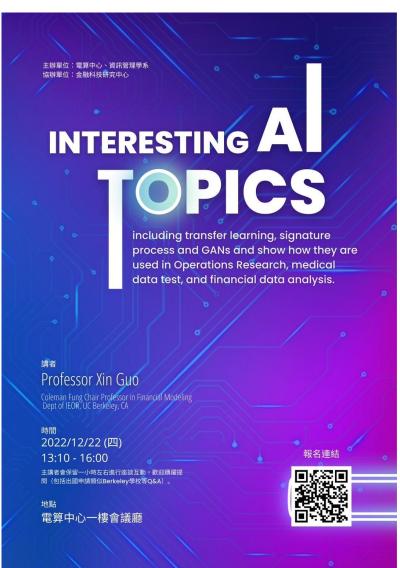
- Write the reflection on12/12 資管專題發表會 or12/22 演講
- File Name:

HW{date_IDnumber}.pdf

ex:

HW1212_110306XXX.pdf HW1222_110306XXX.pdf





Notice

- Remind to send your GitHub link and contact information via Google form https://forms.gle/L1ve3bTjiecEdwNb8
- Keep maintaining your GitHub!
- The make-up section in WM5 will open soon, only can get 4 out of 5 for late homework. The group that didn't upload the proposal to WM5 should also hand-in in the make-up section.