CS-5115 - Programming Prep for Grad Student

PA1 - Data Structures

Arun Totad
WIN: 353146427
Date: 09/24/2024

- I give permission to the instructor to share my solution(s) with the class.
- The main goals of the homework / programming assignment: Create python program to demonstrate how real difference between 1-d Array and Array-List along with obtaining understanding to analyse complexity of the program.

PA1_cs5115_Totad_092424.zip contains the following files:

- PA1_Solution.py – Python code solution for the problem
- Array_Output1.txt & Array_Output2.txt – Gives the output of the program once executed.
- List_Output1.txt & List_Output2.txt – Gives the output of the program once executed.
- Plagiarism declaration.pdf - Plagiarism declaration
- Solution description.pdf – Solution description for the theoretical part of the assignment
- words_alpha.txt – Local dataset for testing purposes

8) Perform a theoretical complexity analysis of your design (i.e., count number of operations/instructions and space usage) and then express that using asymptotic notation as a function of the input size (Pause: what is the input size of your problem?)

Answer:

The input size of the problem will be equal to size of the dataset elements i.e. in this case, n = 370104

As per the code implementation

Time Complexity:

- Loading the dataset: O(n)
- Incremental Growth (Increase Strategy A): O(1)
- Doubling Growth (Increase Strategy B): O(10 * n)
- Fibonacci Growth (Increase Strategy C): O(n)
- Binary Search: Binary search on a sorted list is performed in O(logn). But since insertion dominates: O(n)

Space Complexity

- Loading the dataset: O(n)
- Incremental Growth (Increase Strategy A): O(n)

- Doubling Growth (Increase Strategy B): O(n)
- Fibonacci Growth (Increase Strategy C): O(n)
- Binary Search: Binary search on a sorted list is performed in O(logn). But since insertion dominates: O(n)

Theoretical Complexity Analysis

- **Time Complexity**: The overall time complexity of the program can be summarized as O(n) due to the dominant cost of growing and inserting elements into the array.
- **Space Complexity**: The space complexity is O(n), as the entire dataset is stored and manipulated in memory.

As per the code implementation

Time Complexity:

- Loading the dataset: O(n)
- Incremental Growth (Increase Strategy A): O(n)
- Doubling Growth (Increase Strategy B): O(n)
- Fibonacci Growth (Increase Strategy C): O(n)
- Binary Search: Binary search on a sorted list is performed in O(log n). But since insertion dominates: O(n)

Space Complexity

- Loading the dataset: O(n)
- Incremental Growth (Increase Strategy A): O(n)
- Doubling Growth (Increase Strategy B): O(n)
- Fibonacci Growth (Increase Strategy C): O(n)
- Binary Search: Binary search on a sorted list is performed in O(logn). But since insertion dominates: O(n)

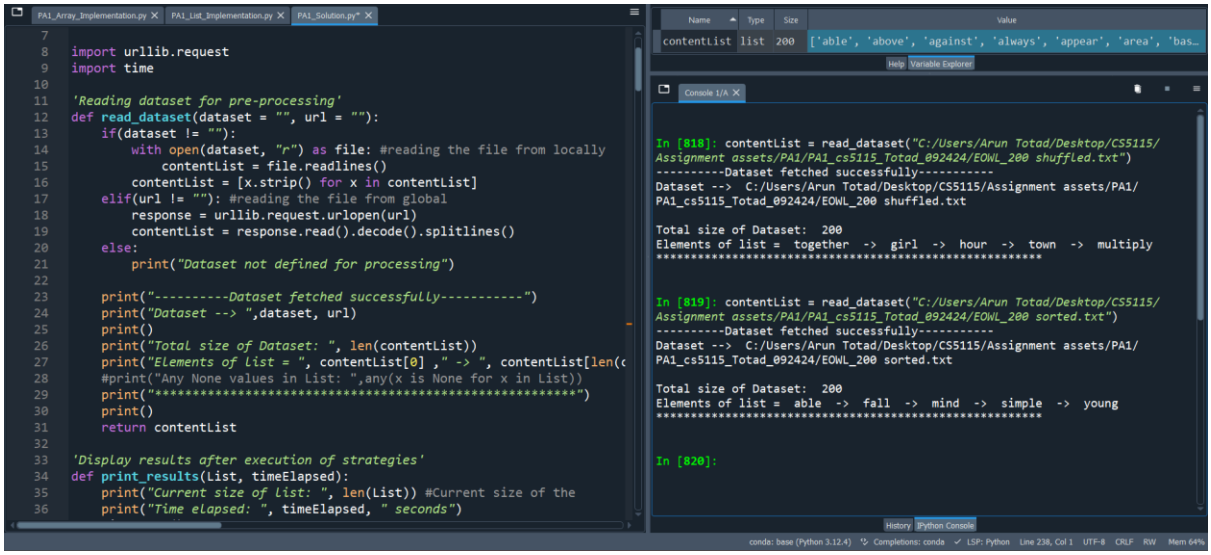
Theoretical Complexity Analysis

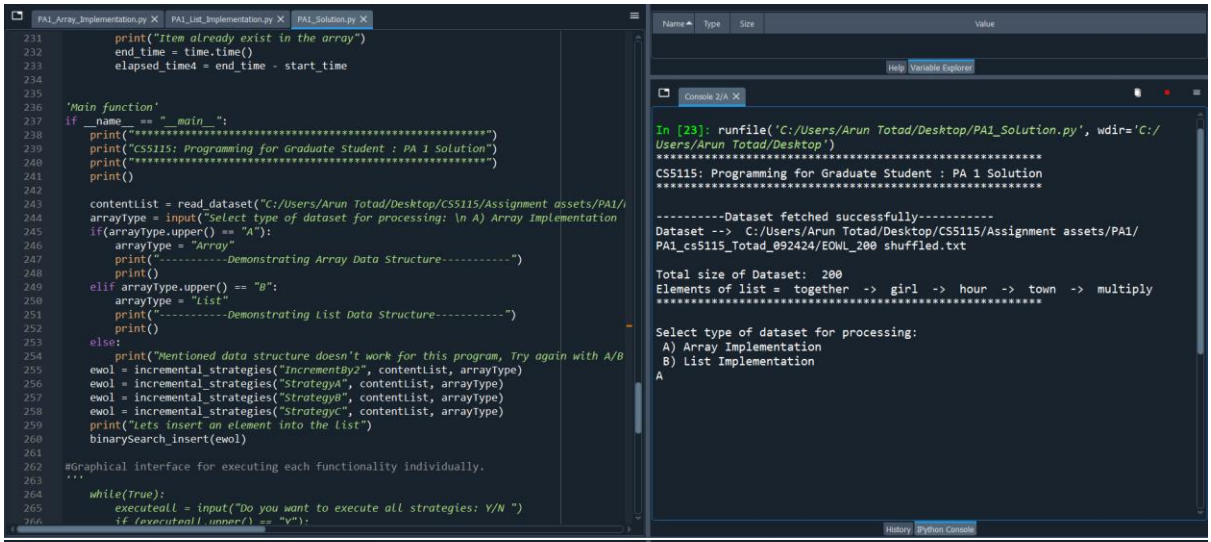
- **Time Complexity**: The overall time complexity of the program can be summarized as O(n) due to the dominant cost of growing and inserting elements into the array.
- **Space Complexity**: The space complexity is O(n), as the entire dataset is stored and manipulated in memory.

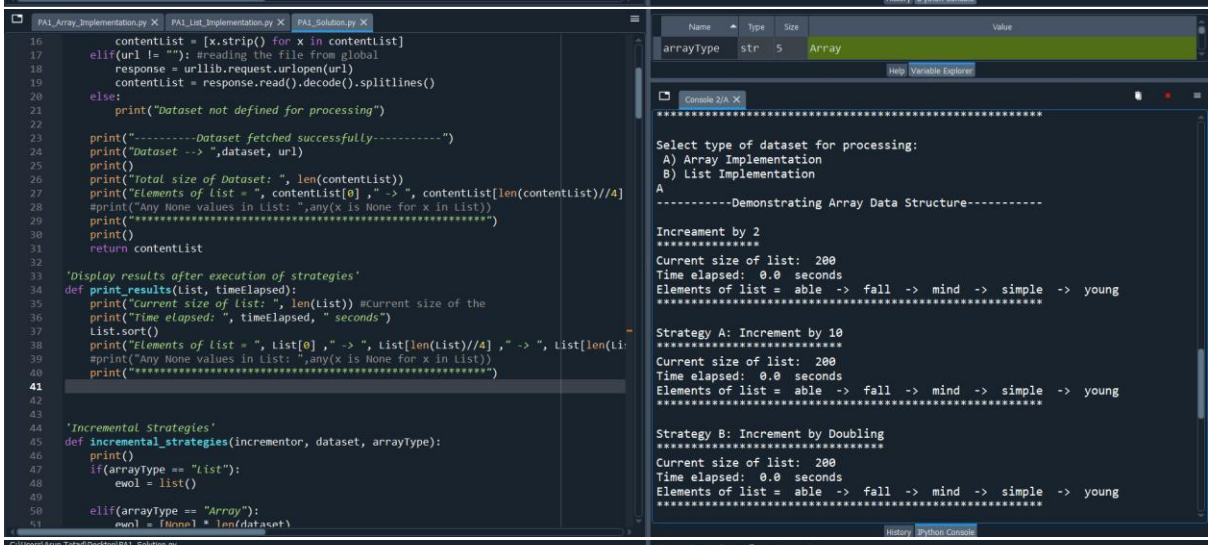
10**.** Empirically measure the time and space complexity of your code

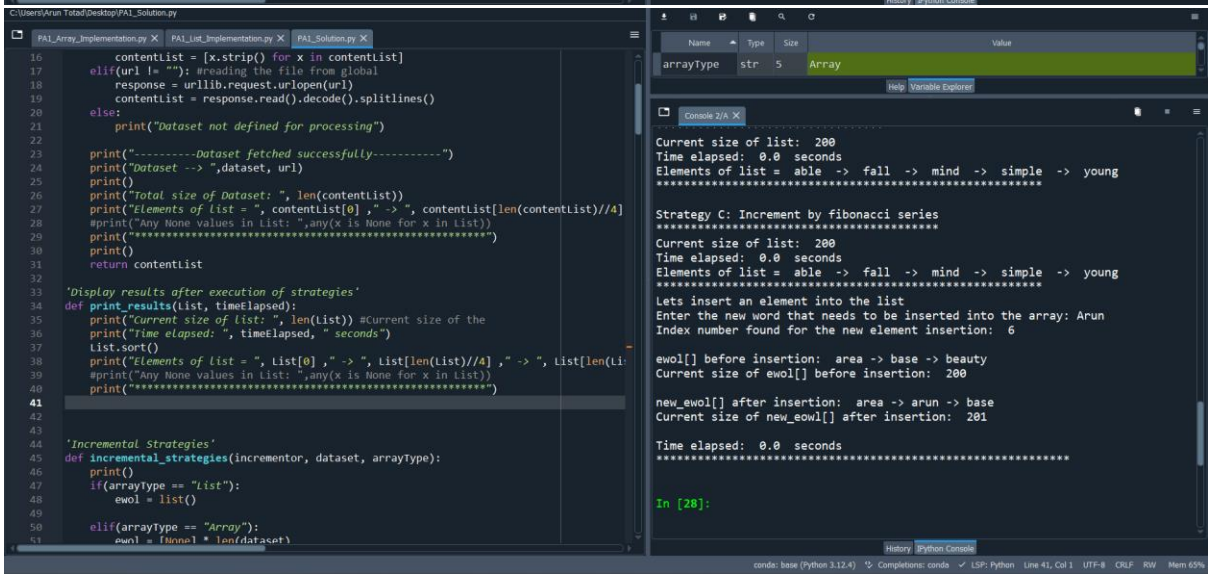
Answer:

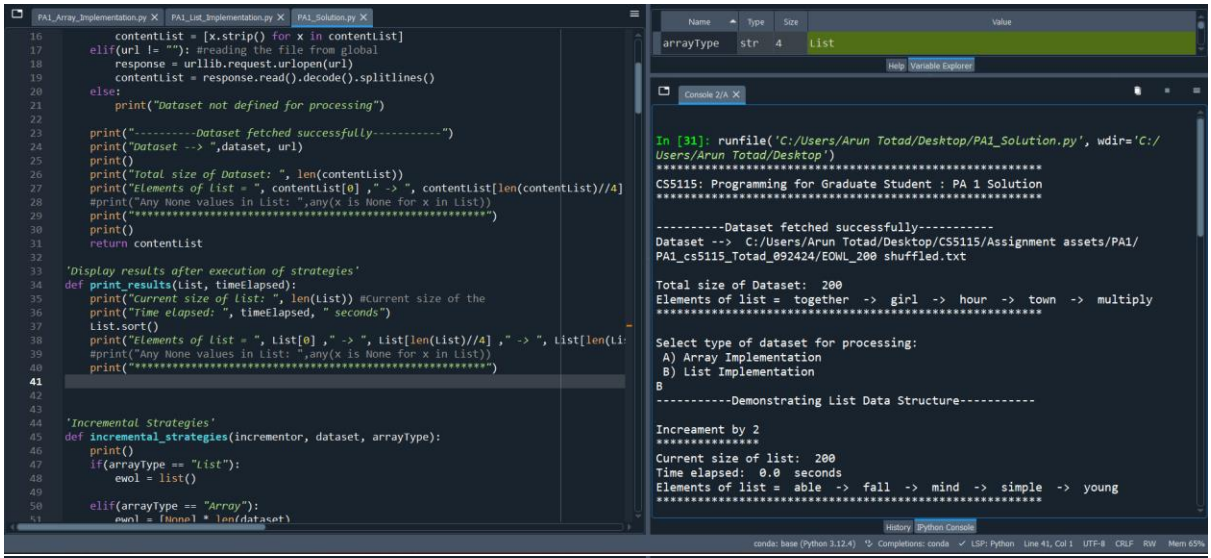
1. **List Implementation**:
    - Final size of the list: 370104 elements.
    - Time for insertion (incrementing by 10 each time): ~0.09377 seconds.
2. **Dynamic Array Implementation**:
    - Final size of the dynamic array: 360102 elements
    - Time for insertion (doubling the array size): ~0.12489 seconds.




PA1_Array_Implementation.py
PA1_List_Implementation.py
PA1_Solution.py
print("Item already exist in the array")
end_time = time.time()
elapsed_time4 = end_time - start_time
'Main function'
if __name__ == "__main__":
print("**********")
print("CS5115: Programming for Graduate Student : PA 1 Solution")
print("**********")
print()
contentList = read_dataset("C:/Users/Arun Totad/Desktop/CS5115/Assignment assets/PA1/
arrayType = input("Select type of dataset for processing: \n A) Array Implementation
if(arrayType.upper() == "A"):
arrayType = "Array"
print("----------Demonstrating Array Data Structure----------")
print()
elif arrayType.upper() == "B":
arrayType = "List"
print("----------Demonstrating List Data Structure----------")
print()
else:
print("Mentioned data structure doesn't work for this program, Try again with A/B
ewol = incremental_strategies("IncrementBy2", contentList, arrayType)
ewol = incremental_strategies("StrategyA", contentList, arrayType)
ewol = incremental_strategies("StrategyB", contentList, arrayType)
ewol = incremental_strategies("StrategyC", contentList, arrayType)
print("Lets insert an element into the list")
binarySearch_insert(ewol)
#Graphical interface for executing each functionality individually.
'''
while(True):
executeall = input("Do you want to execute all strategies: Y/N ")
if (executeall.upper() == "Y"):
Name
Type
Size
Value
Help
Variable Explorer
Console 2/A
In [23]: runfile('C:/Users/Arun Totad/Desktop/PA1_Solution.py', wdir='C:/Users/Arun Totad/Desktop')
****
CS5115: Programming for Graduate Student : PA 1 Solution
****
----------Dataset fetched successfully----------
Dataset --> C:/Users/Arun Totad/Desktop/CS5115/Assignment assets/PA1/PA1_cs5115_Totad_092424/EOWL_200 shuffled.txt
Total size of Dataset: 200
Elements of list = together -> girl -> hour -> town -> multiply
****
Select type of dataset for processing:
A) Array Implementation
B) List Implementation
A
History
IPython Console

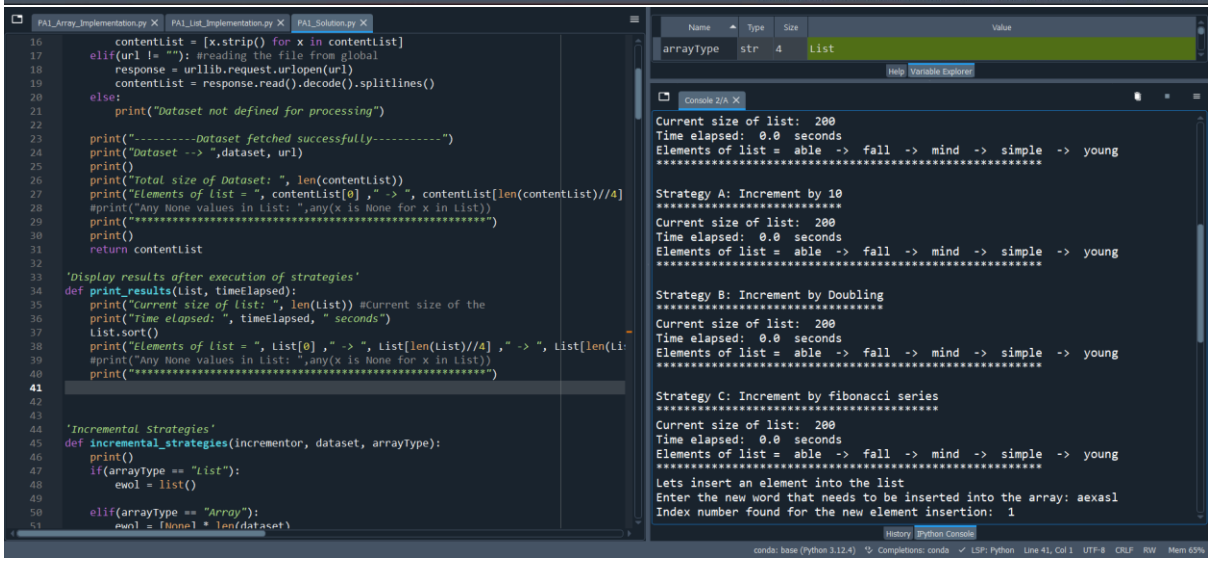

contentList = [x.strip() for x in contentList]
elif(url != ""): #reading the file from global
response = urllib.request.urlopen(url)
contentList = response.read().decode().splitlines()
else:
print("Dataset not defined for processing")
print("----------Dataset fetched successfully----------")
print("Dataset --> ",dataset, url)
print()
print("Total size of Dataset: ", len(contentList))
print("Elements of list = ", contentList[0] ," -> ", contentList[len(contentList)//4]
#print("Any None values in List: ",any(x is None for x in List))
print("**********")
print()
return contentList
'Display results after execution of strategies'
def print_results(List, timeElapsed):
print("Current size of list: ", len(List)) #Current size of the
print("Time elapsed: ", timeElapsed, " seconds")
List.sort()
print("Elements of list = ", List[0] ," -> ", List[len(List)//4] ," -> ", List[len(Li
#print("Any None values in List: ",any(x is None for x in List))
print("**********")
'Incremental Strategies'
def incremental_strategies(incrementor, dataset, arrayType):
print()
if(arrayType == "List"):
ewol = list()
elif(arrayType == "Array"):
ewol = [None] * len(dataset)
Name
Type
Size
Value
arrayType
str
5
Array
Help
Variable Explorer
Console 2/A
****
Select type of dataset for processing:
A) Array Implementation
B) List Implementation
A
----------Demonstrating Array Data Structure----------
Increament by 2
****
Current size of list: 200
Time elapsed: 0.0 seconds
Elements of list = able -> fall -> mind -> simple -> young
****
Strategy A: Increment by 10
****
Current size of list: 200
Time elapsed: 0.0 seconds
Elements of list = able -> fall -> mind -> simple -> young
****
Strategy B: Increment by Doubling
****
Current size of list: 200
Time elapsed: 0.0 seconds
Elements of list = able -> fall -> mind -> simple -> young
****
History
IPython Console


C:\Users\Arun Totad\Desktop\PA1_Solution.py
contentList = [x.strip() for x in contentList]
elif(url != ""): #reading the file from global
response = urllib.request.urlopen(url)
contentList = response.read().decode().splitlines()
else:
print("Dataset not defined for processing")
print("----------Dataset fetched successfully----------")
print("Dataset --> ",dataset, url)
print()
print("Total size of Dataset: ", len(contentList))
print("Elements of list = ", contentList[0] ," -> ", contentList[len(contentList)//4]
#print("Any None values in List: ",any(x is None for x in List))
print("**********")
print()
return contentList
'Display results after execution of strategies'
def print_results(List, timeElapsed):
print("Current size of list: ", len(List)) #Current size of the
print("Time elapsed: ", timeElapsed, " seconds")
List.sort()
print("Elements of list = ", List[0] ," -> ", List[len(List)//4] ," -> ", List[len(Li
#print("Any None values in List: ",any(x is None for x in List))
print("**********")
'Incremental Strategies'
def incremental_strategies(incrementor, dataset, arrayType):
print()
if(arrayType == "List"):
ewol = list()
elif(arrayType == "Array"):
ewol = [None] * len(dataset)
Name
Type
Size
Value
arrayType
str
5
Array
Help
Variable Explorer
Console 2/A
Current size of list: 200
Time elapsed: 0.0 seconds
Elements of list = able -> fall -> mind -> simple -> young
****
Strategy C: Increment by fibonacci series
****
Current size of list: 200
Time elapsed: 0.0 seconds
Elements of list = able -> fall -> mind -> simple -> young
****
Lets insert an element into the list
Enter the new word that needs to be inserted into the array: Arun
Index number found for the new element insertion: 6
ewol[] before insertion: area -> base -> beauty
Current size of ewol[] before insertion: 200
new_ewol[] after insertion: area -> arun -> base
Current size of new_eowl[] after insertion: 201
Time elapsed: 0.0 seconds
****
In [28]:
History
IPython Console
conda: base (Python 3.12.4)
Completions: conda
LSP: Python
Line 41, Col 1
UTF-8
CRLF
RW
Mem 65%



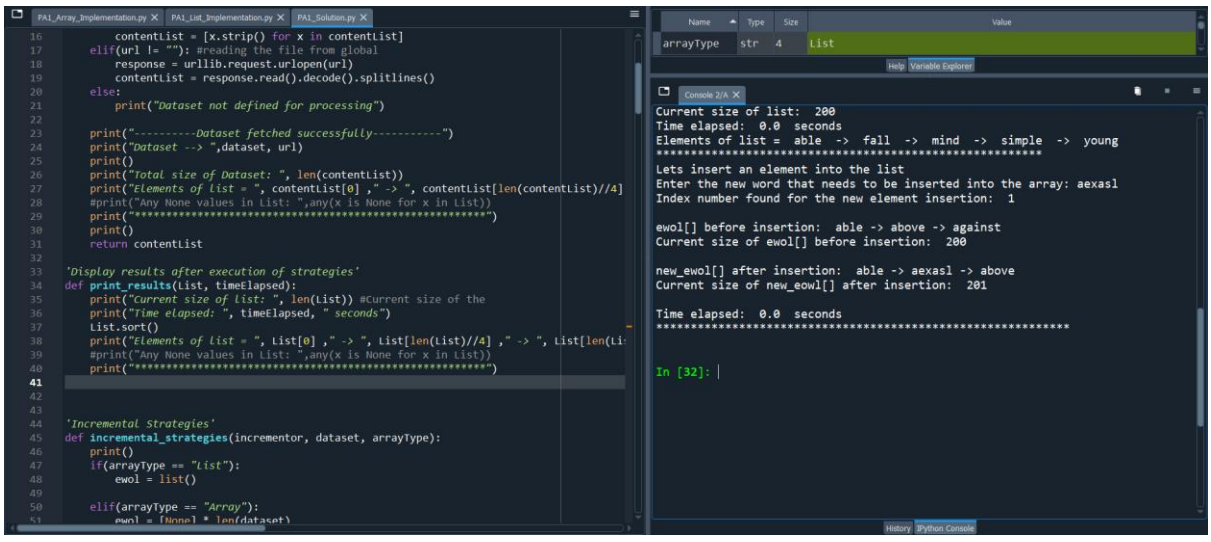
```
        contentList = [x.strip() for x in contentList]
    elif(url != ""): #reading the file from global
        response = urllib.request.urlopen(url)
        contentList = response.read().decode().splitlines()
    else:
        print("Dataset not defined for processing")

    print("----------Dataset fetched successfully----------")
    print("Dataset --> ",dataset, url)
    print()
    print("Total size of Dataset: ", len(contentList))
    print("Elements of list = ", contentList[0] ," -> ", contentList[len(contentList)//4]
    #print("Any None values in List: ",any(x is None for x in List))
    print("*********************************************************")
    print()
    return contentList

'Display results after execution of strategies'
def print_results(List, timeElapsed):
    print("Current size of list: ", len(List)) #Current size of the
    print("Time elapsed: ", timeElapsed, " seconds")
    List.sort()
    print("Elements of list = ", List[0] ," -> ", List[len(List)//4] ," -> ", List[len(Li
    #print("Any None values in List: ",any(x is None for x in List))
    print("*********************************************************")


'Incremental Strategies'
def incremental_strategies(incrementor, dataset, arrayType):
    print()
    if(arrayType == "List"):
        ewol = list()

    elif(arrayType == "Array"):
        ewol = [None] * len(dataset)
```

```
Name        Type    Size    Value
arrayType   str     4       List

Help  Variable Explorer

Console 2/A

In [31]: runfile('C:/Users/Arun Totad/Desktop/PA1_Solution.py', wdir='C:/
Users/Arun Totad/Desktop')
*********************************************************
CS5115: Programming for Graduate Student : PA 1 Solution
*********************************************************

----------Dataset fetched successfully----------
Dataset -->  C:/Users/Arun Totad/Desktop/CS5115/Assignment assets/PA1/
PA1_cs5115_Totad_092424/EOWL_200 shuffled.txt

Total size of Dataset:  200
Elements of list =  together  ->  girl  ->  hour  ->  town  ->  multiply
*********************************************************

Select type of dataset for processing:
 A) Array Implementation
 B) List Implementation
B
----------Demonstrating List Data Structure----------

Increament by 2
**************
Current size of list:  200
Time elapsed:  0.0  seconds
Elements of list =  able  ->  fall  ->  mind  ->  simple  ->  young
*********************************************************

History  IPython Console

conda: base (Python 3.12.4)  Completions: conda  LSP: Python  Line 41, Col 1  UTF-8  CRLF  RW  Mem 65%
```



```
        contentList = [x.strip() for x in contentList]
    elif(url != ""): #reading the file from global
        response = urllib.request.urlopen(url)
        contentList = response.read().decode().splitlines()
    else:
        print("Dataset not defined for processing")

    print("----------Dataset fetched successfully----------")
    print("Dataset --> ",dataset, url)
    print()
    print("Total size of Dataset: ", len(contentList))
    print("Elements of list = ", contentList[0] ," -> ", contentList[len(contentList)//4]
    #print("Any None values in List: ",any(x is None for x in List))
    print("*********************************************************")
    print()
    return contentList

'Display results after execution of strategies'
def print_results(List, timeElapsed):
    print("Current size of list: ", len(List)) #Current size of the
    print("Time elapsed: ", timeElapsed, " seconds")
    List.sort()
    print("Elements of list = ", List[0] ," -> ", List[len(List)//4] ," -> ", List[len(Li
    #print("Any None values in List: ",any(x is None for x in List))
    print("*********************************************************")


'Incremental Strategies'
def incremental_strategies(incrementor, dataset, arrayType):
    print()
    if(arrayType == "List"):
        ewol = list()

    elif(arrayType == "Array"):
        ewol = [None] * len(dataset)
```

```
Name        Type    Size    Value
arrayType   str     4       List

Help  Variable Explorer

Console 2/A

Current size of list:  200
Time elapsed:  0.0  seconds
Elements of list =  able  ->  fall  ->  mind  ->  simple  ->  young
*********************************************************

Strategy A: Increment by 10
**************************
Current size of list:  200
Time elapsed:  0.0  seconds
Elements of list =  able  ->  fall  ->  mind  ->  simple  ->  young
*********************************************************

Strategy B: Increment by Doubling
*********************************
Current size of list:  200
Time elapsed:  0.0  seconds
Elements of list =  able  ->  fall  ->  mind  ->  simple  ->  young
*********************************************************

Strategy C: Increment by fibonacci series
*****************************************
Current size of list:  200
Time elapsed:  0.0  seconds
Elements of list =  able  ->  fall  ->  mind  ->  simple  ->  young
*********************************************************
Lets insert an element into the list
Enter the new word that needs to be inserted into the array: aexasl
Index number found for the new element insertion:  1

History  IPython Console

conda: base (Python 3.12.4)  Completions: conda  LSP: Python  Line 41, Col 1  UTF-8  CRLF  RW  Mem 65%
```



```
        contentList = [x.strip() for x in contentList]
    elif(url != ""): #reading the file from global
        response = urllib.request.urlopen(url)
        contentList = response.read().decode().splitlines()
    else:
        print("Dataset not defined for processing")

    print("----------Dataset fetched successfully----------")
    print("Dataset --> ",dataset, url)
    print()
    print("Total size of Dataset: ", len(contentList))
    print("Elements of list = ", contentList[0] ," -> ", contentList[len(contentList)//4]
    #print("Any None values in List: ",any(x is None for x in List))
    print("*********************************************************")
    print()
    return contentList

'Display results after execution of strategies'
def print_results(List, timeElapsed):
    print("Current size of list: ", len(List)) #Current size of the
    print("Time elapsed: ", timeElapsed, " seconds")
    List.sort()
    print("Elements of list = ", List[0] ," -> ", List[len(List)//4] ," -> ", List[len(Li
    #print("Any None values in List: ",any(x is None for x in List))
    print("*********************************************************")


'Incremental Strategies'
def incremental_strategies(incrementor, dataset, arrayType):
    print()
    if(arrayType == "List"):
        ewol = list()

    elif(arrayType == "Array"):
        ewol = [None] * len(dataset)
```

```
Name        Type    Size    Value
arrayType   str     4       List

Help  Variable Explorer

Console 2/A

Current size of list:  200
Time elapsed:  0.0  seconds
Elements of list =  able  ->  fall  ->  mind  ->  simple  ->  young
*********************************************************
Lets insert an element into the list
Enter the new word that needs to be inserted into the array: aexasl
Index number found for the new element insertion:  1

ewol[] before insertion:  able -> above -> against
Current size of ewol[] before insertion:  200

new_ewol[] after insertion:  able -> aexasl -> above
Current size of new_eowl[] after insertion:  201

Time elapsed:  0.0  seconds
*********************************************************


In [32]:

History  IPython Console
```