

Logbook for ADT

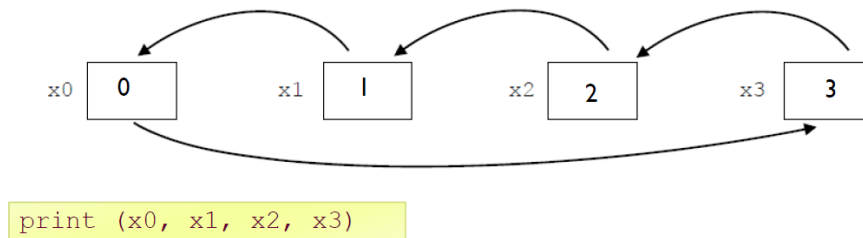
Stefan Ahmed

21359035

Mar 2019

Week 2

1. Suppose that there are 4 variables names x0, x1, x2 and x3. Write the code to move the values stored in those variables to the left, with the leftmost value ending up in the rightmost variable, as shown in the diagram below.



Your answer provided here.

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Fri Feb 22 14:16:59 2019
4
5 @author: 21359035
6 """
7
8 x0 = 0
9 x1 = 1
10 x2 = 2
11 x3 = 3
12
13 y = 0
14
15 x0 = x1
16 x1 = x2
17 x2 = x3
18 x3 = y
19 print(x0, x1, x2, x3)
20
```

2. Write a function that calculates the area of a circle

$$\text{Area} = A = \pi r^2$$

```
21 import math
22 radius = float(input("Enter a radius: "))
23 Area = math.pi * radius * radius
24 print(Area)
25
```

```
Console 1/A
File "C:\ProgramData\Anaconda3\lib\site-packages\spyder_kernels\customize\spydercustomize.py", line 108, in execfile
exec(compile(f.read(), filename, 'exec'), namespace)

File "C:/Users/21359035/Documents/Assignment1.py", line 22, in <module>
    radius = float(input("Enter a radius: "))

ValueError: could not convert string to float:

In [5]:

In [5]: import math
...: radius = float(input("Enter a radius: "))
...: Area = math.pi * radius * radius
...: print(Area)
...:

Enter a radius:
Traceback (most recent call last):

  File "<ipython-input-5-402bf7a76843>", line 2, in <module>
    radius = float(input("Enter a radius: "))

ValueError: could not convert string to float:

In [6]:

In [6]: runfile('C:/Users/21359035/Documents/Assignment1.py', wdir='C:/Users/21359035/Documents')
1 2 3 0

Enter a radius: 10
314.1592653589793

In [7]:
```

3.

Better linear search

```

24 #print(Area)
25
26 def better_linear_search(A, x):
27     for i in range(0, len(A)):
28         if A[i] == x: return print (i)
29     return print(-1)
30
31 better_linear_search([10, 5, 9, 9], 12)
32
33 def better_linear_search(A, x):
34     for i in range(0, len(A)):
35         if A[i] == x: return print(i)
36     return print(-1)
37
38 better_linear_search([10, 5, 9, 9], 9)
39

```

```

30 #
31 #better_linear_search([10, 5, 9, 9], 12)
32
33 def better_linear_search(A, x):
34     for i in range(0, len(A)):
35         if A[i] == x: return print (i)
36     return print(-1)
37
38 better_linear_search([10, 5, 9, 9], 50)
39 |

```

The screenshot shows a Jupyter Notebook console window titled 'Console 1/A'. It displays the execution of the `runfile` function for the file `Assignment1.py` with the working directory set to `C:/Users/21359035/Documents`. The console shows the following sequence of inputs and outputs:

- In [7]: `runfile('C:/Users/21359035/Documents/Assignment1.py', wdir='C:/Users/21359035/Documents')`
- In [8]: `runfile('C:/Users/21359035/Documents/Assignment1.py', wdir='C:/Users/21359035/Documents')` → 0
- In [9]: `runfile('C:/Users/21359035/Documents/Assignment1.py', wdir='C:/Users/21359035/Documents')` → 1
- In [10]: `runfile('C:/Users/21359035/Documents/Assignment1.py', wdir='C:/Users/21359035/Documents')` → 2
- In [11]: `runfile('C:/Users/21359035/Documents/Assignment1.py', wdir='C:/Users/21359035/Documents')` → -1
- In [12]: `runfile('C:/Users/21359035/Documents/Assignment1.py', wdir='C:/Users/21359035/Documents')` → 2
- In [13]: `runfile('C:/Users/21359035/Documents/Assignment1.py', wdir='C:/Users/21359035/Documents')` → 2
- In [14]: `runfile('C:/Users/21359035/Documents/Assignment1.py', wdir='C:/Users/21359035/Documents')` → 0
- In [15]: `runfile('C:/Users/21359035/Documents/Assignment1.py', wdir='C:/Users/21359035/Documents')` → -1
- In [16]:

```

39
40 def better_linear_search(A, x):
41     for i in range(0, len(A)):
42         if A[i] == x: return print (i)
43     return print (-1)
44
45 better_linear_search([10, 5, 9, 9], 8)

```

The screenshot shows a Jupyter Notebook console window with the following content:

```

In [20]: runfile('C:/Users/21359035/Documents/Assignment1.py', wdir='C:/Users/21359035/Documents')
Traceback (most recent call last):

File "C:\ProgramData\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py", line 3267, in run_code
    exec(code_obj, self.user_global_ns, self.user_ns)

File "<ipython-input-20-d5707c83130e>", line 1, in <module>
    runfile('C:/Users/21359035/Documents/Assignment1.py', wdir='C:/Users/21359035/Documents')

File "C:\ProgramData\Anaconda3\lib\site-packages\spyder_kernels\customize\spydercustomize.py", line 704, in runfile
    execfile(filename, namespace)

File "C:\ProgramData\Anaconda3\lib\site-packages\spyder_kernels\customize\spydercustomize.py", line 108, in execfile
    exec(compile(f.read(), filename, 'exec'), namespace)

File "C:/Users/21359035/Documents/Assignment1.py", line 42
    if A[i] == x: return print i
                        ^
SyntaxError: invalid syntax

In [21]:

In [21]: runfile('C:/Users/21359035/Documents/Assignment1.py', wdir='C:/Users/21359035/Documents')
-1

In [22]:

```

Week 3

1. Suppose an algorithm requires following number of operations for a problem of size n . Specify the dominant term(s) having the steepest increase in n . What is $O()$ for each case?

- $7n-2$
- $n^2 - 3n + 10$
- $3n^3 + 20n^2 + 5$
- $1000n + n^2 + 50000$
- $n/3$
- $n \cdot \log n$
- $n^4 + n \cdot \log(n) + 300n^3$

- $0.01n + 0.01n^2$
- $2n + n^{0.5} + 0.5n^{1.25}$

Answer given here

$$7n - 2 = O(n)$$

$$N^2 - 3n + 10 = O(n^2)$$

$$3n^3 + 20n^2 + 5 = O(n^3)$$

$$1000n + n^2 + 50000 = O(n^2)$$

$$n/3 = O(1) \text{ a constant}$$

$$n \cdot \log n = O(n \log n)$$

$$n^4 + n \cdot \log n + 300n^3 = O(n \log n)$$

$$0.01n + 0.01n^2 = O(n^2)$$

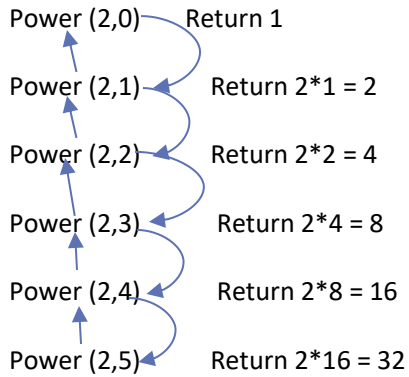
$$2n + n^{0.5} + 0.5n^{1.25} = O(n^{0.5})$$

2. What is the time complexity of the following two algorithms? Express your answer in terms of Big-O notation and justify the answer.

Your answer provided here.

Week 4 – Recursion

1.



3.)

```
def moveDisk(fp,tp):
    print("moving disk from",fp,"to",tp)

def moveTower(n,fromPole, toPole, withPole):
    if n >= 1:
        moveTower(n-1,fromPole,withPole,toPole)
        moveDisk(fromPole,toPole)
        moveTower(n-1,withPole,toPole,fromPole)
```

```
Tower_of_Hanoi.py - C:\Users\21359035.STUDENT.000\Downloads\Tower_of_Hanoi.py (3.5.2)
File Edit Format Run Options Window Help
def moveDisk(fp,tp):
    print("moving disk from",fp,"to",tp)

def moveTower(n,fromPole, toPole, withPole):
    if n >= 1:
        moveTower(n-1,fromPole,withPole,toPole)
        moveDisk(fromPole,toPole)
        moveTower(n-1,withPole,toPole,fromPole)

moveTower(4,"T1","T2","T3")
```

Logbook for ADT

...

```
===== RESTART: C:\Users\21359035.STUDENT.000\Downloads\Tower_of_Hanoi.py =====
moving disk from T1 to T3
moving disk from T1 to T2
moving disk from T3 to T2
moving disk from T1 to T3
moving disk from T2 to T1
moving disk from T2 to T3
moving disk from T1 to T3
moving disk from T1 to T2
moving disk from T3 to T2
moving disk from T3 to T1
moving disk from T2 to T1
moving disk from T3 to T2
moving disk from T1 to T3
moving disk from T1 to T2
moving disk from T3 to T2
>>>
```

Week 5 – Stacks

1.)

C. 'z' is the top item on the list when the sequence is complete.

4.) Evaluating postfix expressions

$$(2*3)+4 = 10$$

$$2*(3+4) = 14$$

$$(1*3)*(2+4)$$