

BT 3051 — Data Structures and Algorithms for Biology

Jul–Nov 2016

Quiz 1

September 8, 2016

Name: I. VIJAY

Roll number: BE13B015

Instructions: This quiz is 'closed book' — but you can use only *your* hand-written class notes. Answer all questions. **Keep your answers brief and to the point.** Allotted time is 50 minutes.

There are a total of 7 pages in this quiz.

Total marks: 50

1. (8 marks) **Justify** (as briefly as possible) whether each of the following statements is true or false. If the justification is incorrect, no credit will be awarded.

- (a) `[7].__mul__(1)` will return the value `[7]`

True

It is of the form `a.__mul__(b)` will give the result `a*b`

$\Rightarrow [7] * (1)$

$\Rightarrow [7]$

$[a] * n$
 $= [a \dots n \text{ times}]$

- (b) The following snippet of code will *never* return a zero value for values of $\delta \in (0, 1)$:

```
def test_float_add_sub(delta):  
    x = 0  
  
    for i in range(10):  
        x += delta  
    for i in range(10):  
        x -= delta  
  
    return x
```

False

For the case of $0.1 \in (0, 1)$ the output is not zero

For all other values $\in (0, 1)$ the output is zero

Because float point 0.1 is a special case

- (c) Tuples of strings are permissible keys for dictionaries in Python

True

In defining a dictionary the elements is of the

form of `TupleSeq('a')`

(d) The following snippet of code will produce an error:

```
a = list(range(10,0,-1))
b = a.sort()
print(b)
```

True

list indexing should start from '0'

a = list(range(0,10,-1))

2. (12 marks) Indicate the correct answer(s) for the following questions with multiple choices and explain your answer. Partial credit will not be awarded.

(a) Consider a file ComplexNumberFile.py with the following code:

```
class ComplexNumber:
    def __init__(self, r, i):
        self._real = r
        self._imag = i

print(__name__)
```

If you then run a file ImportTest.py containing the following code:

```
from ComplexNumberFile import ComplexNumber
```

the output will be:

i. __main__ ii. ComplexNumber ☒ iii. ComplexNumberFile iv. ImportTest

It will import ComplexNumber class, But the complex Number class has no item to print, It asked for print(__name__) which will call for main function that is ComplexNumber File

(b) Which of the following are possible states of an array of numbers [6, 3, 2, 4, 5, 8, 7, 1] during an insertion sort (in ascending order):

☒ i. [3, 6, 2, 4, 5, 8, 7, 1] ☒ ii. [2, 3, 4, 6, 5, 8, 7, 1] ☒ iii. [2, 3, 6, 4, 5, 8, 7, 1] iv. [1, 2, 3, 4, 6, 5, 8, 7]

6, 3, 2, 4, 5, 8, 7, 1

3, 6, 2, 4, 5, 8, 7, 1 ✓

2, 3, 6, 4, 5, 8, 7, 1 ✓

2, 3, 4, 6, 5, 8, 7, 1 ✓

2, 3, 4, 5, 6, 8, 7, 1

2, 3, 4, 5, 6, 8, 7, 1

2, 3, 4, 5, 6, 7, 8, 1

1, 2, 3, 4, 5, 6, 7, 8

(c) Recall ArrayQueue discussed in class. If the following code is run,

```
from ArrayQueue import *
Q = ArrayQueue()
for x in range(10):
    Q.enqueue(x)
for x in range(5):
    dummy = Q.dequeue()
```

the final state of Q._data is (Assume DEFAULT_CAPACITY = 10):

- i. [5, 6, 7, 8, 9]
- ✓ ii. [5, 6, 7, 8, 9, None, None, None, None, None]
- ✓ iii. [None, None, None, None, None, 5, 6, 7, 8, 9]
- iv. [None, None, None, None, None, None, None, None, None, None, None, None, None, None, None, 5, 6, 7, 8, 9]

for x in range(10): pushes 10 elements from 0 to 10
Q.enqueue(x) → creates a memory of 10 elements

for x in range(5): pops
Q.dequeue() → pops 5 numbers into the memory out of list

Final state is [5, 6, 7, 8, 9, None, None, None, None, None]

(d) The snippet of code shown below:

```
def Sort(a):
    n = len(a)
    for i in range(n - 1):
        swapped = False
        for j in range(n - 1 - i):
            if a[j] >= a[j + 1]:
                a[j], a[j + 1] = a[j + 1], a[j]
                swapped = True
        if not swapped:
            break
    return a
```

- ✓ i. sorts 'a' in ascending order
- ii. sorts 'a' in descending order
- ✓ iii. is adaptive
- ✓ iv. is stable

which is an implementation of bubble sort, which sorts
a in ascending order

→ which is stable & adaptive

Section II

3. (3 marks) What is the output of the following snippet of code? Explain your answer.

```
#!/usr/bin/python3
# -*- coding: utf-8 -*-

f = -1
g = 2
for i in range(8):
    print(f)
    f+=g
    g=-g
    g+=f
```

-1
1
0
1
1
2
3
5

Reason:-

$f = f + g$ $f = -1 + 2 = 1$ ✓ Now $f = 1$
 $g = -g$ $g = -1$ ✓
 $g = g + f$ $g = -1 + 1 = 0$ ✓
 and then it continues to give

4. (3 marks) Spot and correct the error in the following snippet of code, which tries to reverse a number:

```
#!/usr/bin/python3
# -*- coding: utf-8 -*-

n = 123456789
m = 0
for i in range(len(str(n))):
    m = 10*m + (n % 10)
    n = n//10

print(m,n)
```

In the line `for i in range(len(str(n))):` \rightarrow `str(n)` is defined
 But `n` is not defined as string \rightarrow which is an error

Correct:-

`n = '123456789'`

`m = 0`

`for i in range(len(str(n))):`

`m = 10*m + (n % 10)`

`n = n//10`

5. (4 marks) Explain with an example array of integers, as to why selection sort is not stable.

Selection sort is not stable, Because $\Theta(n^2)$ comparisons, $\Theta(n)$ swaps

Example:-

8 2 3 1 1
 1 2 3 1 8
 1 1 3 2 8
 1 1 2 3 8

In this example even if there are same numbers
 it will still do comparisons and swaps
 Hence not stable

6. (4 marks) What happens when the following program is run?

```
with open('nums.txt') as f:
    x = f.readlines()
    print(x)
    for line in x:
        print(line.strip())
        z = f.readline()
        print(z)

print(f.readlines())
```

nums.txt looks like this:

```
1
2
3
4
5
```

$x = f.readlines()$ → reads for the lines in nums.txt

and prints → $[1\n, 2\n, 3\n, 4\n, 5\n]$

next it will strip each line and prints

$[1, 2, 3, 4, 5]$

$z = f.readline()$ → will read first line and prints $'1'$

OUT PUT

$[1\n, 2\n, 3\n, 4\n, 5\n]$

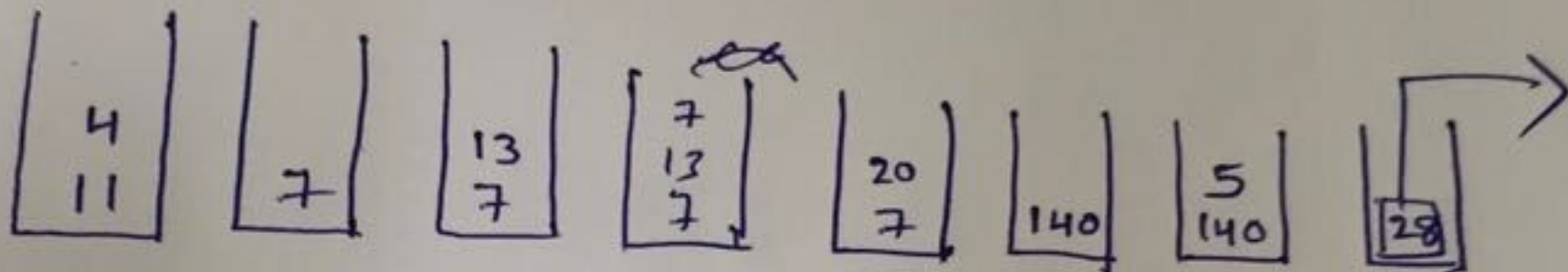
$[1, 2, 3, 4, 5]$

1

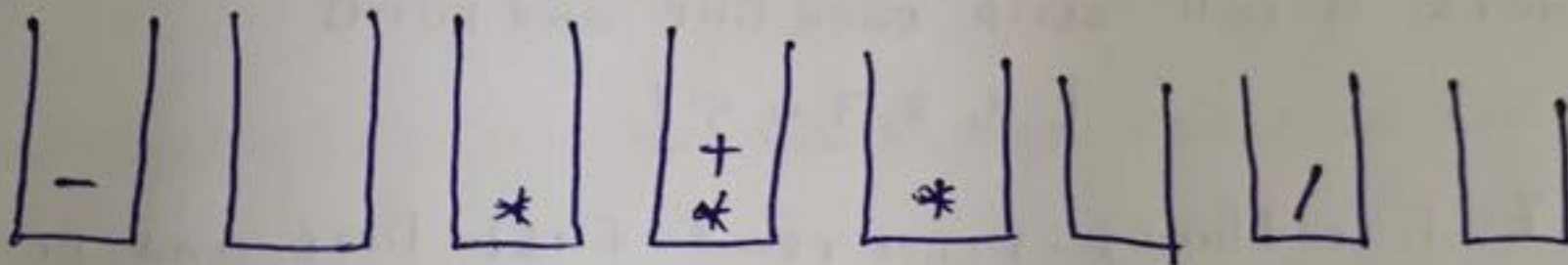
7. (6 marks) For the fully parenthesised expression " $((11-4)*(13+7))/5$ ", trace the growth of the operand and operator stacks, as the expression is completely evaluated using Dijkstra's two-stack algorithm.

$$((11-4)*(13+7))/5$$

operand



operator



output is 28

$$\begin{array}{r} 4, 11 \\ 11-4 \end{array}$$

$$\begin{array}{r} 7+13 \\ 20 \end{array}$$

$$\begin{array}{r} 5) 140(28 \\ 10 \\ \hline 40 \end{array}$$

8. (10 marks) Define a minimalist Rational class to represent rational numbers, such that the following statements, which access relevant class methods, return values as shown:

```
>>> print(Rational(10, 20))
1/2
>>> print(Rational(10, 20) + Rational(30, 20))
2
```

Assume that a gcd function, which computes the greatest common divisor of two integers, is already written for you and is available as gcd().

```
class Rational:
```

```
    def __init__(self, n, d=1):
```

```
        self.n = n
```

```
        self.d = d
```

```
    def simplify(self):
```

```
        x = gcd(self.n, self.d)    // gcd() is available
```

```
        self.n = self.n / x
```

```
        self.d = self.d / x
```

```
        print(Rational(self.n, self.d))
```

```
    def __str__(self):
```

```
        return str(self.n) + "/" + str(self.d)
```

```
    def __add__(self, other):
```

```
        return Rational(self.n * other.d + other.n * self.d, self.d + other.d)
```

```
if __name__ == '__main__':
```

```
    print print(Rational(10, 20))
```

```
    print(Rational(10, 20) + Rational(30, 20))
```

*** END OF QUIZ 1 ***