Name Chng Soon Siang

TFIP - Milestone 2 Quiz

70 +2

72

14 December 2020

3 hours

Additional Material(s): Blank papers for workings

Instructions to candidates:

- Read all instructions and questions carefully.
- Do not open this examination paper until instructed to do so.
- Write legibly in dark blue or black pen.
- Do not use staples, paper clips, glue or correction fluid.

- Write your name at the top of each page, including the papers used for workings.
- Answer all the questions.
- Write your answers in the space provided.
- You are to submit all documents, including papers used for workings.
- You are reminded not to bring any documents out of the examination room.

- This document consists of 17 printed pages.
- The number of marks is given in brackets [ ] at each question.
- The maximum number of marks for this paper is **100**.

Name: Chng Soon Siang

1. Arrays and Linked Lists [6 marks]

You have been tasked to choose between an array or a linked list data structure for storing a large data set. You have also been given the following information:

- **new entries are constantly appended to the data set**
- **size of each entry of data is not fixed (i.e. some entries may have more information than others while some entries may have less)**

Which data structure will you choose? In terms of space and time efficiencies, explain your decision of choosing one over the other.

Answer:

Choose linked-list.

~~Reason~~: Considerations:

~~1) Array has a maximum size~~

~~1) Array needs to be initialize~~

1) A linked-list ~~has insertion of O(1) while it~~ is able to store data dynamically, the size and the type of data would not matter whereas an array would need to be re-assigned memory each time the size of the array changes or the data within it is changed. ✓

2) Insertion to a linked-list ✗ will take O(n) since ~~it~~ pointer has to traverse to the end to ~~insert~~ append the new ~~a~~ node. ✓ will be O(1) if a tail pointer is maintained.

3) Linked-list requires more ~~less~~ memory but memory allocation is dynamic. ✓

Conclusion:

The requirement for the capability to store data dynamically ~~outweighs the disadv~~ is only satisfied by linked-list and it outweighs the disadvantage of O(n) insertation time complexity.

5

Name : Ong Soon Siang

2. FizzBuzz [10 marks]

Write a program that prints each number 1 to 100 on a new line. However, there are several exceptions:

- For each multiples of 3, the programs prints "Fizz" instead of the number
- For each multiples of 5, the program prints "Buzz" instead of the number
- For numbers that are both a multiple of 3 and 5, the program prints "FizzBuzz" (or "Fizz Buzz") instead of the number

Output of the first 15 values:

```
1
2
Fizz
4
Buzz
Fizz
7
8
Fizz
Buzz
11
Fizz
13
14
FizzBuzz    # "Fizz Buzz" is also accepted
```

Write your answers in the next page.

Name: Ching Soon Siang

Answer:

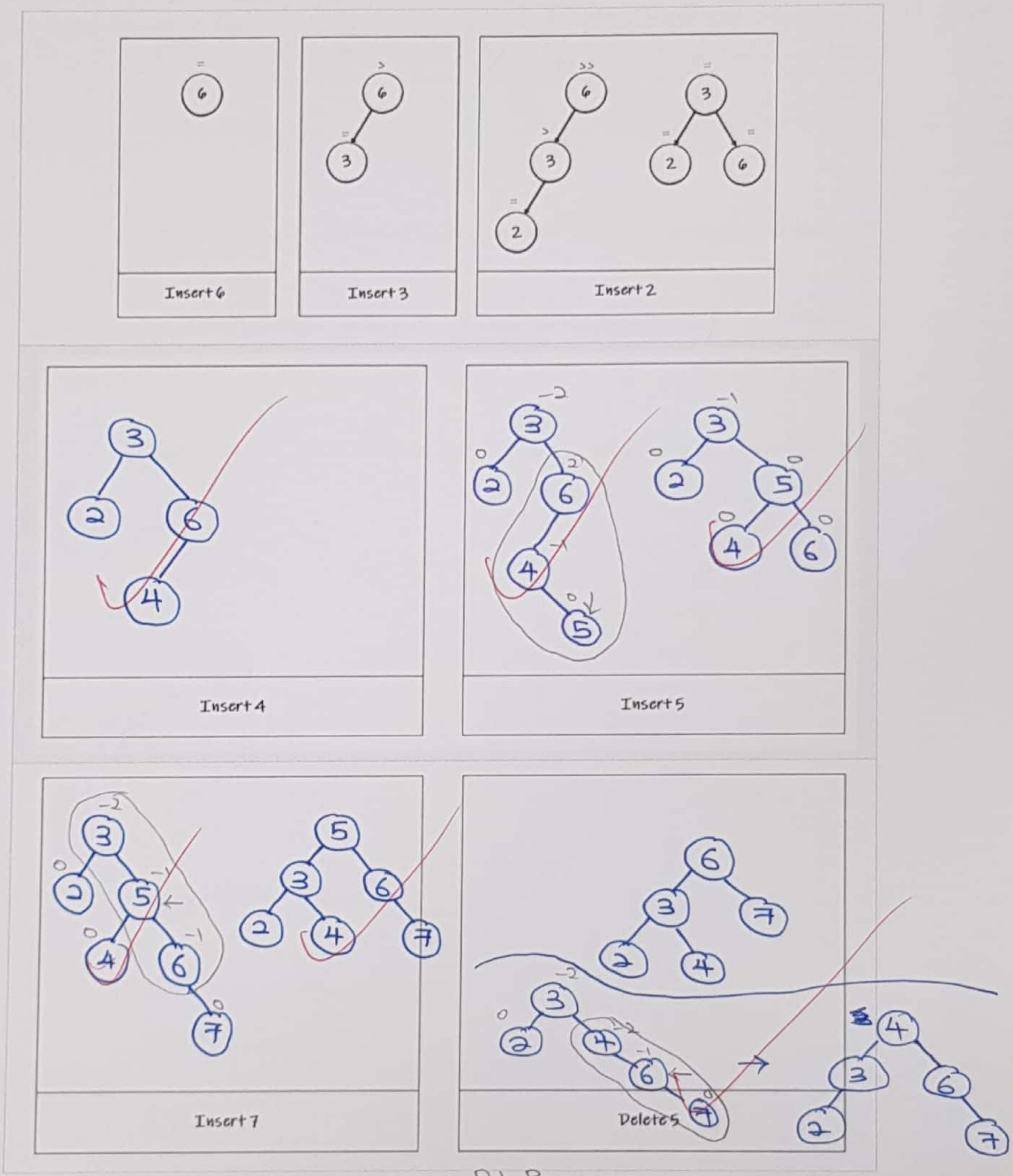
```
for num in range(1,101,1):
    if num%3==0 and num%5==0:
        print("FizzBuzz")
    elif num%3==0:
        print("Fizz")
    elif num%5==0:
        print("Buzz")
    else:
        print(num)
```

10

Name: Ching Soon Siang

3. Trees [14 marks]

(a) You are to construct the AVL tree based on the following inputs. For each input, if there is any balancing required, draw the AVL tree before and after balancing. (9 marks)



RLR

a

Name: Ong Soon Siang

(b) Based on the results in part (a), give the Preorder, Inorder and Postorder traversals of the AVL Tree? (3 marks)

↓
Ro L R

Preorder: 6, 3, 2, 4, 7

Inorder: 2, 3, 4, 6, 7

↓
L R Ro

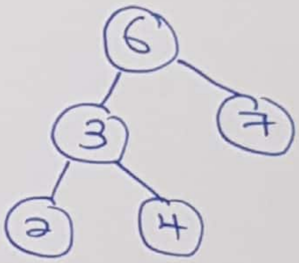
Postorder: 2, 4, 3, 7, 6

(c) What is the height of the AVL Tree in part (a)? (1 mark)

Height: 3

(d) What is the big O notation of the `insert` operation in AVL Trees? (1 mark)

O (log n)



Root left right
6, 3, 2, 4, 7
Left right root
2, 4, 3, 7, 6

3

Name: Chng Soon Siang

4. Functional Programming [20 marks]

Write an algorithm for the custom function `significant_numerals()` for the `map()` function that does the following:

- removes all ~~non~~ leading alphabet or symbols from the numeral strings
- removes all leading zeros from the numeral strings
- you may use of the functions `find()` and `isdigit()`. Their descriptions from the Python documentation are provided below:
    - `find()` - Return the lowest index in the string where substring *sub* is found within the slice `s[start:end]`. Optional arguments *start* and *end* are interpreted as in slice notation. Return `-1` if *sub* is not found.
    - `isdigit()` - Return `True` if all characters in the string are digits and there is at least one character, `False` otherwise.
    - `strip()`, `lstrip()` & `rstrip()` - Return a copy of the string with the leading and/or trailing characters removed
- assume that the input lists consists of `Strings` datatypes only
- regular expressions is NOT required!

Use the following codes to help you:

```
def significant_numerals(str_num):
    # TODO: fill in the function

currency_list = ['EUR000058649', '$845', 'USD0000548584', '€000595',
'$00000.25']
print(list(map(significant_numerals, currency_list)))
# Results: ['58649', '845', '548584', '595', '0.25']
```

Write your answers on the next page.

Name: Chng Soon Siang

```
def significant_numerals(str_num):
    ~~While isdigit~~ done = True
    while done:
        idx = find(str-num, ~~[illegible]~~ start=0, end=-1) ?
        if idx != -1:
            str-num = str-num[idx:]
        elif isdigit(str-num):
            return ~~str-num~~ lstrip(str-num, 0)
                                str-num.lstrip ('0')
```

4

Name: Chng Soon Siang

5. Object-Oriented Programming [18 marks]

Consider the following Python code and answer the questions on the next page.

```
class Spell:
    def __init__(self, incantation, name):
        self.name = name
        self.incantation = incantation
    def __str__(self):
        return f'{self.name}, {self.incantation}\n{self.get_description()}'
    def get_description(self):
        return 'No description'
    def execute(self):
        print(self.incantation)

class Accio(Spell):
    def __init__(self):              → incantation
        Spell.__init__(self, 'Accio', 'Summoning Charm')
                                                 ↳ name

class Confundo(Spell):
    def __init__(self):             incantation        name
                                         ↑                ↑
        Spell.__init__(self, 'Confundo', 'Confundus Charm')
    def get_description(self):
        return 'Causes the victim to become confused and befuddled.'

def study_spell(spell):
    print(spell)

spell = Accio()
spell.execute() → print 'Accio'
study_spell(spell)
study_spell(Confundo())
```

Name: Chng Soon Siang

(a) What are the parent and child classes here? (3 marks)

Parent: Spell

Child: Accio, Confundo,

(b) What does the code print out? (5 marks)

```
Accio
Summoning Charm, Accio
No description
```

~~<[illegible] [illegible] [illegible]>~~

```
Confundus Charm, Confundo
Causes the victim to be confused and befuddled.
```

(c) Which `get_description()` method is called when `study_spell(Confundo())` is executed?
Why? (4 marks)

The one that returns the string 'Causes the victim...' because of method overriding, it replaces the method of the parent class.

(d) What do we need to do such that `print(Accio())` will print the description: `'This charm summons an object to the caster, potentially over a significant distance'` ? Write down the code that we need to add and/or change. (6 marks)

X -4

```
def __str__(self):
    return 'This charm summons an object to the caster, potentially over a significant distance'
```

this works but breaks OOP principles.

14

t.2

Name: Chng Soon Siang

(b) Answer the following questions. (2 marks)

(i) Starting at vertex A, write the order at which the vertices are visited when traversing the graph using depth-first search (DFS).

Note: Adjacent vertices are to be visited in alphabetical order.

(ii) Is DFS implemented using a Stack or a Queue?

Answer:

(i) A → B → D → F → H → C → E → G

(ii) Stack ✓

(c) Answer the following questions. (2 marks)

(i) Starting at vertex A, write the order at which the vertices are visited when traversing the graph using breadth-first search (BFS).

Note: Adjacent vertices are to be visited in alphabetical order.

(ii) Is BFS implemented using a Stack or a Queue?

Answer:

(i) A → B → C → D → E → F → G → H

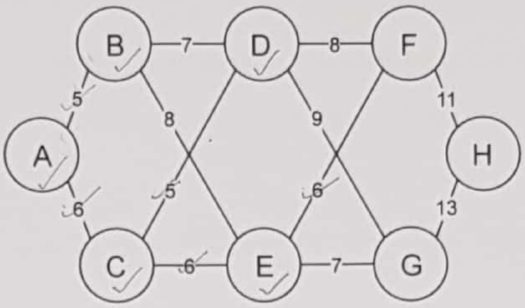
(ii) Queue ✓

3

Name: Chng Soon Siang

6. Graphs [12 marks]

For the following questions, refer to the graph below. You are not expected to implement any Python code.



8 nodes
12 edges

(a) Construct the adjacency matrix of the graph above. (4 marks)

Answer:

	A	B	C	D	E	F	G	H
A	0	5	6	0	0	0	0	0
B	5	0	0	7	8	0	0	0
C	6	0	0	5	6	0	0	0
D	0	7	5	0	0	8	9	0
E	0	8	6	0	0	6	7	0
F	0	0	0	8	6	0	0	11
G	0	0	0	9	7	0	0	13
H	0	0	0	0	0	11	13	0

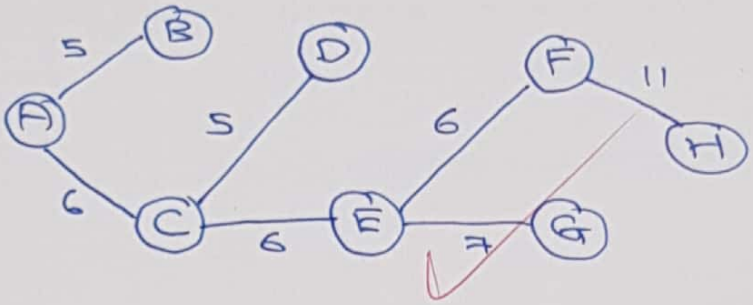
✓

4

Name: Chng Soon Siang

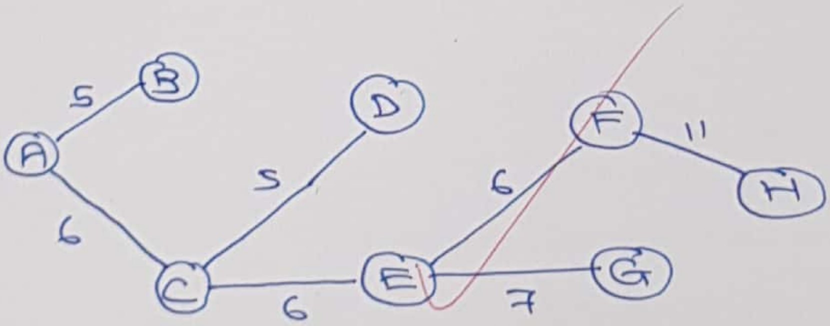
(d) Starting at vertex A, draw the resulting graph using Prim's algorithm. (2 marks)

Answer:



(e) Starting at vertex A, draw the resulting graph using Kruskal's algorithm. (2 marks)

Answer:



✓(A,B,5)
✓(A,C,6)
×(B,D,7)
×(B,E,8)
✓(C,D,5)
✓(C,E,6)
×(D,F,8)
~~(E,G,9)~~
✓(F,H,11)
×(G,H,13)

✓(E,F,6)
×(D,G,9)
✓(E,G,7)

4

Name : Chng Soon Siang

7. Code Debugging [20 marks]

The following questions involve Stock prices.

As an additional note:

> To compute the percentage change in stock price, simply subtract the previous closing price from the current price and divide the difference by the previous closing price. Then, multiply by 100 to get the percent change. If the sign is negative, that means that the price decreased.

(a) Identify and correct the errors in the code snippet below. (16 marks).

```
Stock:
    def __init__(symbol, name)
        self.symbol = symbol
        self.name = abc
        self.prev_closing_price = 0
        self.__current_price
        return

    def load_from_data_record(cls, data_record):
        temp = cls(data_record, data_record)
        temp.set_urrent_price(data_record)
        temp.set_prev_closing_price(data_record)

    def get_change_percent(self):
        ((self.get_current_price() -
         self.get_prev_closing_price())/self.get_prev_closing_price()) * 100

    def set_current_price(self, curr_price):
        self.__current_price = curr_price
    def get_current_price(self):
        return self.__current_price

    def set_prev_closing_price(self, close_price):
        close_price
    def get_prev_closing_price(self):
        return self.prev_closing_price

    def __str__(self):
        out = f'Symbol: {self.symbol}\n'
        out += f'Name: {self.name}\n'
        out += f'Current Price: {self.get_current_price()}'
        return out
```

Name: Chng Soon Siang

Code that runs it (Note: there are no errors in the code snippet below):

```
# Code that runs it:
data = [
    {'symbol':'INTC', 'name':'Intel Corp', 'current_price':51.99,
'prev_closing_price':50.99},
    {'symbol':'AMD', 'name':'Adv Micro Devices', 'current_price':94.04,
'prev_closing_price':92.31},
    {'symbol':'GOOG', 'name':'Alphabet Cl C', 'current_price':1828,
'prev_closing_price':1827},
    {'symbol':'BA', 'name':'Boeing Co', 'current_price':232.71,
'prev_closing_price':237.20},
    {'symbol':'ORCL', 'name':'Oracle Corporation', 'current_price':59.96,
'prev_closing_price':59.27}
]
     dict   list
for item in data:                        dict
    s = Stock.load_from_data_record(item)
    print(s)
    print(f'Previous Closing Price: {s.get_prev_closing_price()}')
    print(f'Percentage Change: {s.get_change_percent():.2f}\n')
```

(a) Answer:

class Stock:

    def --init--(self, symbol, name, current-price, prev-closing-price):

        self.symbol = symbol

        self.name = name

        --

        self.prev-closing-price=0   -1

        0

        self.--current-price = current-price   -1

    @classmethod.   -1   cls   -1

    ~~cls~~ self

    def load-from-data-record(~~self~~, data-record):

        ~~temp = cls(datarecord~~

        self.symbol = data-record['symbol']

Name: Chng Soon Siang

```
self.name=data-record['name']
self.set-current-price(data-record['current-
                                   price'])

~~self~~ self.set-prev-closing-price(data-record
      ['prev-closing-price'])

def get-change-percent(self):
    return ((self.get-current-price()-
    self.get-prev-closing-price())/
    self.get-prev-closing-price())* 100

def set-current-price(self, curr-price):
    self.--current_price = curr-price
def get-current-price(self):
    return self.--current-price

def set-prev-closing-price(self, close-price):
  ~~retu~~ self.prev-closing-price = close-price
def get-prev_closing-price(self):
   return self.prev_closing-price.

def __str__(self):
   out = f'Symbol: {self.symbol}\n'
   out += f'Name: {self.name}\n'
   out += f'Current Price: {self.get-current-
                            ~~price~~ price()}'
   return out
```

11

-- prev-closing-price should be private

Name: Ching Soon Siang

(b) What is the output of the 3rd stock i.e. GOOG (4 marks)?

(b) Answer:

Symbol: GOOG

Name: Alphabet Cl C x

Current Price: 1828

Previous Closing Price: 1827 ✓

Percent Change: some number with 2 decimal places. ? -1

3