CSC 120 EXAM 1 REVIEW GUIDE

Exam: Friday February 25, 2022
Review Session: Wednesday February 23, 2022. 5:30-7:00pm @ BIO W 301

TOPIC LIST:

Python review

References

Linked Lists

Classes and Objects

WHERE TO STUDY:

- Slide-decks
- ICAs
- This study guide

QUESTIONS?

- Attend the Review Session
- Discord
- Office Hours

Note: these are not necessarily the questions that will be on your test. These questions were written collectively by the TAs (who haven't seen the test yet)! These questions make up a good summary of the course so far, but do not cover every detail that we have gone over in class. Because of this, **this is not intended to be your only resource for studying.**
That being said, feel free to skip questions, and do the ones you think will help you the most. Ask questions on Discord, and attend a review session for additional clarification.

REVIEW PROBLEMS:

1. True or False:

   A) Linked lists can be indexed

   B) An array and a linked list are the exact same thing

   C) Arrays have built in methods that are useful in using and/or manipulating them

   D) .splice() is a method that you can use to separate strings into elements of an array

   E) Linked Lists do not have built in methods

   F) You can access a string without a reference to it

   G) We can copy elements of an array or string with slicing

2. Given a linked list of integers, write a function min_max_list() that accepts the linked list as a parameter and returns a tuple containing the minimum and maximum values of the list. If the list is empty, return None.

   Ex:
   Given the list 0 -> 1 -> -2 -> 4 -> -9
   Your function should return the tuple (-9, 4)

3. Given a two dimensional array, create a function `add_one(array)` which increments each element in the two dimensional array by one

   Example:

   Given the array: [[7,5,6], [3,2,1], [10,11,12,13]]

   Your function should return: [[8, 6, 7], [4, 3, 2], [11, 12, 13, 14]]

4. What is the potential error in this code?

```
cur = head
while cur.next is not None:
cur = cur.next
```

5. Consider the following code:

```
x = “foo”
y = “bar”
z = x

x = “baz”
y = x + y + z

print(x)
print(y)
print(z)
```

What is the output of the program?

Hint: Drawing a reference diagram may be helpful

6. You are writing a program that will keep track of student grades in CSC 120. It must be able to accept an input of a student’s name and print a number that represents their current grade. Which of the following data types would be the best choice for a program of this nature and why?

A) String

B) Array

C) Set

D) Dictionary

E) Linked List

7. Which of the following data structures and data types are considered mutable in Python?

A) Strings

B) Arrays

C) Sets

D) Integers

E) Booleans

F) Tuples

G) Linked Lists

8. What would be the output when we run the following code?

```
arr1 = [1, 2, 3, 4, 5, 6]
arr2 = arr1
for i in range(len(arr2)):
     arr2[i] = arr2[i]*2
arr1[0] = arr2[1]
print(“Array 1:”)
print(arr1)
print(“Array 2:”)
print(arr2)
```

9. Write a class that makes a Turtle whose size, color, age and shell pattern as attributes. You must also add a `birthday(self)` method which increases its age by one and a `grow(self, amount)` that increases the turtle’s size by the specified amount. You must also add getters for all the above attributes.

10. How would you keep track of a tail and why would tails be useful in a linked list?

11. Let’s say you have an object that represents a course at a university. This course has a name, level, instructor, and list of students. Fill in the blank

```
class Course:
      def __init__(self, name, level, instructor):
            ‘‘‘Sets the name, level, and instructor of this course to
            the values provided. Leave the students as an empty list
            Params:
                  name: the name of the course, ex: CSC
                  level: the numerical level of the course, ex: 120
                  instructor: the instructor for the course, ex: Russ
            Returns: no return’’’
            # — part a —

      def add_student(self, new_student):
            ‘‘‘This function adds new_student to the list of students
            Params:
                  new_student: the student to be added
            Returns: no return’’’
            # — b —

      def __str__(self) :
            ‘‘‘This function returns a string with all relevant
            information for the course in the following format:
            “CSC 120 with Russ Lewis has 222 students”
            ** Use an f string **
            Params: no params
            Returns: a string with information about the course’’’
            # — c —
```

12. What would this segment of code print out?

```
def foo(arr):
      ret_val = set()
      for x in arr:
            if x % 3 != 0:
                  ret_val.add(x * 10 + x % 10)
      return ret_val

print(foo([1,2,3,4,5,6]))
```

13. Write a function named `read_val()` that reads input from the user until the user enters “done” and you have to return all the inputs as an array.

    Example:

    Enter some input and type done to exit.

    20

    40

    Hello

    Bye

    Done

    You should return: [“20”, “40”, “Hello”, “Bye”]

14. Write a function named `get_middle(head)` that returns the middle value of the linked list given from the head parameter. If there are two middle elements, return the one on the left. If head is None, return -1.

15. Write a function named `make_2D_array(arr, n)` which takes an array and a number as its parameters and returns a 2D array such that the number of elements in the inner array are the same as n. If there is less than said number you should return that inner array with however many elements it can fit such that all the elements of the 1D array are in the 2D array.

    For Example:

    arr = [1,2,3,4,5,6,7,8,9,10]
    n = 3

    You should return: [[1,2,3], [4,5,6], [7,8,9], [10]]

16. What does the assignment operator (=) do in Python?

17. Draw a reference diagram at each point in the code.
    Hint: Remember that the slicing operation creates an entirely new object

```
x = [None, None, None, None, None]
y = [None, None, None, None]
# DRAW HERE

for i in range(4):
    x[i] = f"hello{i+1}"
# DRAW HERE

for i in range(3):
    y[i] = x[i:i+2]
# DRAW HERE

y[-1] = [x[-1], y[-1], y]
# DRAW HERE
```

18. Create the function `reverse_list(arr)` which takes an array and returns a reversed linked list with the array contents. You may assume that the linked list uses the `ListNode` class we used in class, you do not need to rewrite it.

    For Example:

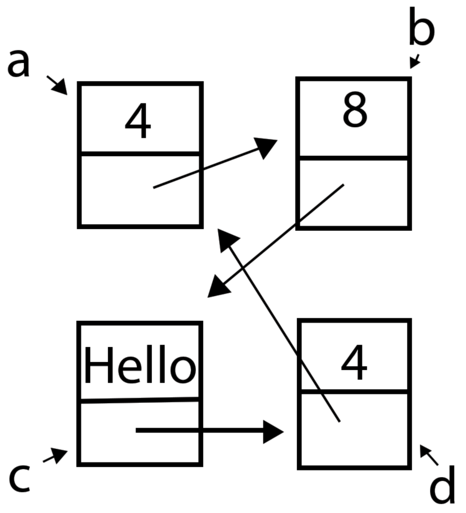
        arr= [1, 2, 3, 4]

        You should return: 4 -> 3 -> 2 -> 1

19. You are talking with your friend about one of your recent programming assignments where you were given various colors and had to combine them to make other colors. Your friend held each color in an array while you used a class to store each color. Both of you did well on the assignment so you wonder, “Why are classes helpful when I could have used an array instead?”

    Give some reasons for why classes are useful.

20. Given the following reference diagram:



Substituting which letters (a ,b, c, or d) for “___” would make the following python statement true:

a. a[1][1][1][1] is ___

b. a[1][1][1][1] == ___

21. Under what circumstances would it be more beneficial to use a linked list over an array? What about an array over a linked list?

22. Are you able to traverse a linked list backwards? Explain your answer.

Solutions

1.
   A) False

   B) False

   C) True

   D) False

   E) True

   F) False

   G) True

2.

```
def min_max_list(head):
      if head is None:
            return None
      min_val = head.val
      max_val = head.val
      cur = head
      while cur is not None:
            if cur.val > max_val:
                  max_val = cur.val
            if cur.val < min.val:
                  min_val = cur.val
            cur = cur.next
      return (min_val, max_val)
```

3.

```
def add_one(array):
      for i in range(len(array)):
            for j in range(len(array[i])):
                  array[i][j] += 1
      return array
```

4. If head = None, then cur.next would produce an AttributeError that None has no attribute next.

5.

```
baz
bazbarfoo
foo
```

6. D - A dictionary is the best choice because of its ability to map a key to a value, in this case a string and an int.

7. B, C, and G are correct answers. The values in arrays, sets, and linked lists can be modified, while the values of strings, integers, booleans, and tuples cannot.

8.

```
Array 1:
[4,4,6,8,10,12]
Array 2:
[4,4,6,8,10,12]
```

9.

```
class Turtle:
      def __init__(self, size, age, color):
            self.size = size
            self.color = color
            self.age = age

      def birthday(self):
            self.age+=1

      def get_age(self):
            return self.age

      def get_color(self):
            return self.age

      def get_size(self):
            return self.age

      def grow(self, amount):
            self.size += amount
```

10. To keep track of a tail pointer, whenever a new node is added to the end of the list, you can set the tail pointer to this new node. It is useful for inserting nodes at the end of the linked list without having to loop through the whole list.

11.

```
a. self._name = name
   self._level = level
   self._instructor = instructor
   self._students = []
```

```
b. self._students.append(new_student)
```

c.

```
return f“{self._name} {self._level} with {self._instructor} has
{len(self._students)} students”
```

12.

{11, 44, 22, 55} (order here doesn’t matter)

13.

```
def read_val():
      print(“Enter some input and type done to exit”)
      flag = True
      arr = []

      while flag:
            user_input = input()

            # to exit the infinite loop
            if user_input == “done”:
                  flag = False
            else:
                  arr.append(user_input)

      return arr
```

14.

This is one solution, but there is a better space-complexity solution!

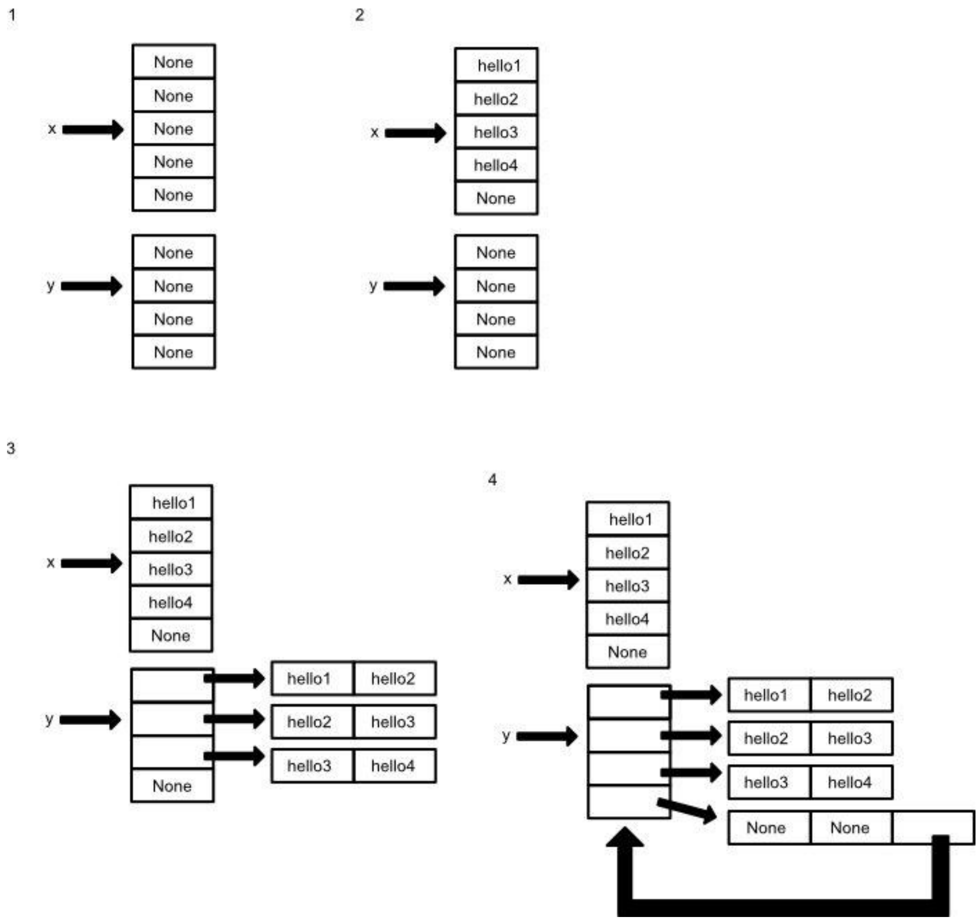
```
def get_middle(head):
  if head is None:
        return -1
  elements = []
  cur = head
  while cur:
        elements.append(cur.val)
        cur = cur.next
  if len(elements)%2==0:
        # if list is even there are two middles
        return elements[(len(elements)//2)-1]
  else:
        return elements[len(elements)//2]
```

15. One possible solution:

```
def make_2D_array(arr,n):
    arr_2D = []

    for i in range(0,len(arr),3):
        inner_arr = []

        for j in range(0,n):

            # checking if the array index goes out of range or not
            if (i+j) < len(arr):
                inner_arr.append(arr[i+j])

        arr_2D.append(inner_arr)

    return arr_2D
```

16. The assignment operator first evaluates the right hand side. It figures out what object is on the right and then makes the variable on the left reference that object on the right.

17.



18.

```
def reverse_list(arr):
    if len(arr) == 0:
        return None

    head = ListNode(arr[0])
    if len(arr) == 1:
        return head

    for val in arr[1:]:
        new_head = ListNode(val)
        new_head.next = head
        head = new_head

    return head
```

19. Classes provide abstraction. It allows us to think of the problem at a higher level. Classes allow us to use name fields instead of having to worry about which index what color is in (some people may do [R, G, B] where others may want [G, R, B], etc.). Furthermore, classes can give us the freedom to change our internal representation of the data. As long as our method outputs stay the same, any code that relies on our class will still work. If we were using the arrays, it would be much harder to do this. Classes can allow us to easily accept only valid colors. With a class, we could throw an error in our setter and constructor when the color is invalid. With the array, it's much harder to do as you may have to duplicate your checks across all the functions that may accept the array.

20.
   a) a
   b) a

21. When you have a dataset that needs to have insertions at the front, linked lists are faster at doing those insertions. When an item needs to be instantly indexed at a known position, an array would be more beneficial since it is not necessary to loop through the linked list to find the element.

22. You cannot technically traverse a linked list backwards. In order to reverse the elements of a linked list, you would need to create an entirely new linked list with a different ordered collection of elements.