## Data Science C - Data Science C - UT Invitational 2020 - Division C - 11-08-2020

This exam has two parts: 1) A set of coding challenges hosted on Google Colab. 2) A test on Scilympiad.

For the coding challenges, be sure to read all instructions carefully (failure to do so may result in losing points).

This trial event was originated by Dhruva Karkada in the 2019-2020 season and was held for the first time at the UT Austin Invitational. We're excited to be running it again!

## 1. (40.00 pts)

The Google Colab notebook containing the coding challenges can be found here: https://colab.research.google.com/drive/1iWSIJ4LJx4lyPNMNihc\_Apdvclbl--ef?usp=sharing (https://colab.research.google.com/drive/1iWSIJ4LJx4lyPNMNihc\_Apdvclbl--ef?usp=sharing)

This is the master copy. It will not let you save the file if you edit it. To complete the challenges, **click the "Copy to Drive" button in the upper left**. This will create a private copy of this file in your google drive, which you can edit and save. Make only one copy per team -- your partner should be able to edit the notebook simultaneously, as long as you both have access to the document

access to the document.	
To submit your notebook, make sure to enable link sharing. This is done by clicking the "Share" button in the upper right, clicking "Change to anyone with the link" at the bottom of the popup, and then clicking "Copy link." Make sure that you share with the setting "Anyone on the internet with this link can EDIT." This is extremely important, as it allows us to check if you edited your code after the submission time. If you do not do share your Colab Notebook with this setting, your coding challenges will not be scored.	
Before your test ends on Scilympiad, you should submit your Google Colab link through this google form: FORM LINK (https://forms.gle/Nq7Qs9e71QEsVFzNA)	

```
1 from enum import Enum
2
3 class Role(Enum):
4
     TAMK = 0
5
     DAMAGE = 1
6
     SUPPORT = 2
8 class Hero:
9
    ult_charge = 0
10
11
     def __init__(self):
         pass
12
1.3
    def add ult charge(self, amount):
14
15
        self.ult_charge += amount;
16
17
     def print_info(self):
        print("I'm a hero in the game Overwatch!");
18
19
20
21 class Overwatch (Hero):
22
    def init (self):
23
        pass
24
25
    def print info(self):
        print("I'm part of Overwatch");
26
27
28 class Talon(Hero):
    def __init__(self):
29
30
         pass
31
32
    def print info(self):
33
        print("I'm part of Talon!");
34
35 class Human(Hero):
36
    def __init__(self):
37
        pass
38
39
     def print_info(self):
        print("I'm a human!");
40
41
42 class Tracer(Overwatch, Human):
43
44
     hp = 150
     role = Role.DAMAGE
45
46
    needed_charge = 25
47
48
     def __init__(self):
49
         pass
50
51
     def cast_ult(self, stick=False):
52
        if(self.ult_charge):
53
            if(stick):
54
                 print("You need a time out!")
55
              else:
56
                 print("Bombs away")
57
58
             print("My ultimate is charging")
59
60 class Moira(Talon, Human):
hp = 200
    role = Role.SUPPORT
63
    needed_charge = 100
64
65
      def __init__(self):
66
         pass
67
68
    def cast_ult(self):
69
       if(self.ult_charge):
70
             print("Géill do mo thoil!")
71
         else:
72
            print("My ultimate is charging")
73
74
75 def main():
76
    t = Tracer();
77
      m = Moira();
79 if __name__ == "__main__":
    main()
80
```

2. (1.00 pts)	Which of the following lines exhibits variable instantiation?
○ A) 23	
○ В) 9	
O C) 40	
O D) 52	
3. (1.00 pts)	Which of the following lines exhibits multiple inheritance?
O A) 42	
○ в) 35	
O C) 76	
O D) 1	
4. (1.00 pts)	Which of the following lines exhibits singular inheritance?
4. (1.00 pts)	Which of the following lines exhibits singular inficinance:
O A) 25	
○ В) 15	
O C) 60	
O D) 21	
5. (1.00 pts)	Which of the following lines exhibits assignment of a default argument?
O A) 48	
○ в) 51	
O C) 15	
O D) 5	
6. (1.00 pts)	Which of the following lines exhibits method overriding?
	· · · · · · · · · · · · · · · · · · ·
O A) 77	
○ В) 14	
O C) 11	
○ D) 25	
7. (1.00 pts)	Which of the following lines exhibits creation of an enum?
O A) 80	
О в) 77	
○ C) 3	
O D) 1	
8. (1.00 pts)	Which of the following lines exhibits instantiation of an object?
	Which of the following lines exhibits instantiation of an object?
3. (1.00 pts)  A) 76  B) 3	Which of the following lines exhibits instantiation of an object?

P. (1.40 pts)   Experience we created a Tracer ediptot and called print into method. What wexter pietro out?   A. Impart of Cheevedath     S. Impart of Taker!     C. D. My ultimate is charging	OD) 6				
<ul> <li>○ g) Fin a humani</li> <li>○ g) Impart of Tailori</li> <li>○ D) My utilized is obserging</li> <li>10. (1.00 pts) What is the main difference between the methods _ssand _repr_?</li> <li>○ A) _repr_ is the human-readable format, while _ssis a more technical format.</li> <li>○ B) There is no difference between the xwapp_ is a logge feature time hythor 2</li> <li>○ G) _repr_ is between the xwapp_ is a logge feature time hythor 2</li> <li>○ D) _resr_ returns a user if dendly representation the object, while _repr_ generally veltures the raw object format</li> <li>11. (1.00 pts) Suppose I had a Pythor Statement such as the following:</li></ul>	9. (1.00 pts)	Suppose we created a Tracer object and called print info method. What would print out?			
C c) Pro Part of Boord  D) My ultimate is changing  10. (1.00 pts) What is the main difference between the methods _sband _repr_?  (A) _repris the human-readable format, while _stis a more technical format  C B) There is no difference between the two, _repr is a legacy feature from Python 2  C c) _reprshould only be called by the option, while _stic and be called by owen  C C) _reprshould only be called by the option, while _stic and be called by owen  D) _sbreturns a user intendity representation the object, while _reprgenerally returns the raw object format  11. (1.00 pts) Suppose that a Python Statement such as the following:  [b to x in regist(d, ib) st x ii ii z = ii]  This would be an example of what python language corcept?  A) List comprehension  C c) List reduction  C p) For loop family the fail in the previous question would contain when?  A) [0. 2, 4, 6]  C) [1.3, 5, 7]  D) [1.3, 5, 7, 9]  13. (1.00 pts) What is the average time complexity of preparching to a system list?  A) O(1)  C) (0) (0)  C) (00 pts)  Vivid is the average time complexity of preparching to a system list?  A) (1.100 pts) What is the average time complexity of preparching in advanced from a dictionary?  A) (0.11  C) (0.00 pts)  Vivid is the average time complexity of preparching in a demonst from a dictionary?  A) (0.11  C) (0.00 pts)  D) (0.00 pts)  C) (0.00 pts)  D) (0.00 pts)	O A) I'm part	of Overwatch			
D (), My ultimate is changing  10, (1.00 ptb) What's the main difference between the methods _ str_ and _ repr_?  A) _ repr_ is the human-readable format, while _ str_ is a more technical format  B) Time to mail difference between the two repr_ is a legacy feature from Python 2  C) _ repr_ should only be called by the system, while _ str_ can be called by uses  C) _ par_ returns a user friendly representation the object. while _ str_ can be called by uses  11, (1.00 ptb) Suppose I had a Python Statement auto as the following:  [x for x in reage(0, 18) ztr x 5 2 → 8]  This would be an example of what python language concept?  A) List comprehension  D) For loop transformation  12, (1.00 ptb) The list in the previous question would contain what?  A) [12, 4.60 ptb) The list in the previous question would contain what?  A) [12, 4.6 ptb) The list in the previous question would contain what?  A) [13, 4.6 ptb) What is the average time composity of prepending to a python list?  A) O(1)  B) O(n)  C) O(n)  C) O(n)  D) O(n)  D) O(n)  D) O(n)  15, (1.00 ptb) Suppose I had an ordered list of numbers of size n and i wanted to insert them into a Binary Search Tree? What would be the big-O of his operation?	○ B) I'm a hu	uman!			
10. (1.00 pts) What is the resen difference between the methods _str_ and _repr_2  A \ _repr_ is the human-seadable format, while _str_ is a more technical format  B \ There is no difference between the low _ repr_ is a legate feature from Python 2  C _ repr_ debated only be colliced by the system, while _str_ can be colled by users  D _ str_ returns a user thereby representation the object, while _repr_ generally returns the raw object format  11. (1.00 pts) Suppose these a Python Statement such as the following:  [ x for x in respect, 10] if x x 2 = 6]  This would be an example of what jrython language concept?  A \ Lat comprehension  B \ Lat comprehension  C \ Lat refunction  C \ D \ The log in the previous question would contain what?  A \ D \ The log in the previous question would contain what?  A \ D \ The log in the previous question would contain what?  A \ D \ The log in the previous question would contain what?  A \ D \ The log in the everage time complexity of prepareding to a python list?  A \ C \ D \ This would be an everage time complexity of prepareding to a python list?  A \ C \ D \ D \ D \ D \ D \ D \ D \ D \ D	O C) I'm part	of Talon!			
A)expr_ is the human-readeble format whilesr is a more technical format  ② b) There is no difference between the two,exp is a legacy feature from Python 2  ③ c)expr should only be called by the system, whilest can be called by uses  ③ c)expr should only be called by the system, whilest can be called by uses  ③ c)exp should only be called by the system, whilest can be called by uses  ① d)exp should be an example of what python is not be following:  [x for x to respect(a, its) if x x 2 8]  This would be an example of what python is nguisge consest?  A) List comprehension  ② b) List preneation  ② c) List reduction  ① p For loop transformation  12 (1.00 pts) The list in the previous question would contain what?  A) (0, 2, 4, 6)  ② b) (0, 2, 4, 6, 6)  ③ b) (0, 2, 4, 6, 6)  ③ b) (0, 2, 4, 6, 6)  ③ c) (1, 3, 5, 7, 9)  13 (1.00 pts) What is the average time complexity of propending to a python list?  A) O(1)  ⑤ c) O(n²)  ⑤ p) O(log n)  14 (1.00 pts) What is the average time complexity of rotificing an element from a dictionary?  A) O(c)  ⑤ p) O(n²)  D) O(n²)	O D) My ultin	nate is charging			
<ul> <li>□ g) There is no difference between the two, _ropr_ is a legacy feature from Python 2</li> <li>□ c)escshould only be called by the system, whitestrcan be called by users</li> <li>□ b)strreturns a user friendly representation the object, whileroprgenerally returns the raw object format</li> <li>11. (1.00 pts) Suppose I had a Python Statement such as the following:  [x for x to ranget(9, 16) 1f x x 2 -= 6]  This would be an example of what python language concept?</li> <li>○ A) List comprehenation</li> <li>□ B) List generation</li> <li>□ C) List reduction</li> <li>□ C) For loop transformation</li> <li>12. (1.00 pts) The fat is the previous question would contain what?</li> <li>○ A) [0, 2, 4, 6]</li> <li>□ B) [0, 2, 4, 6]</li> <li>□ D) [1, 3, 5, 7]</li> <li>□ D) [1, 3, 5, 7]</li> <li>□ D) [1, 3, 5, 7, 8]</li> <li>13. (1.00 pts) What is the average time complexity of prepending to a python list?</li> <li>○ A) O(1)</li> <li>□ B) O(n)</li> <li>□ C) O(n/2)</li> <li>□ D) O(this n)</li> <li>14. (1.00 pts) What is the average time complexity of retrieving an element from a dictornary?</li> <li>○ A) O(1)</li> <li>□ B) O(this n)</li> <li>□ C) O(n)</li> <li>□ D) O(this n)</li> <li>□ D) O(this</li></ul>	10. (1.00 pts)	What's the main difference between the methodsstr andrepr?			
C C) _reprshould only be called by the system, while _str_ can be called by users  11. (1.00 pts) Supcose I had a Python Statement such as the following:  [x for x in range(0, 1a) if x x 2 - a]  This would be an example of what python language concept?  A) List comprehension  C) List reduction  D) For loop transformation  12. (1.00 pts) The list in the previous question would contain what?  A) (0.2, 4, 6]  B) (0.2, 4, 6]  C) [1.3, 5, 7]  D) (1.3, 5, 7, 9]  13. (1.00 pts) What is the average time complexity of preponding to a python list?  A) O(1)  C) O(n*2)  D) O(neg n)  14. (1.00 pts) What is the average lime complexity of retrieving an element from a dictionary?  A) O(1)  C) O(n*2)  C) O(n*3)  C) O(n*4)  C) O(n*	O A)repr_	_ is the human-readable format, whilestr is a more technical format			
ti. (1.00 pts) Suppose I had a Python Statement such as the following:  [x for x in range(e, 18) if x x 2 = e] This would be an example of what python language concept?  A ) List comprehension B ) List generation C ) List reduction D ) For loop transformation  12 (1.00 pts) The list in the previous question would contain what?  A ) (0, 2, 4, 6] B ) (0, 2, 4, 6, 8) C ) (1, 3, 5, 7) D ) (1, 3, 5, 7, 9)  13 (1.00 pts) What is the average time complexity of prepending to a python list?  14 (1.00 pts) What is the average time complexity of retrieving an element from a dictionary?  A ) (0, 1) B ) (0, 2) B ) (0, 3) B ) (0, 3) B ) (0, 4) B ) (0, 4) B ) (0, 5) B ) (0, 6) B ) (0, 7)	O B) There is	s no difference between the two,repr is a legacy feature from Python 2			
11. (1.00 pts) Suppose I had a Python Statement such as the following:  [x for x in range(0, 16) if x x 2 = 0]  This would be an example of what python language concept?  A just comprehension  B just generation  C just reduction  D j For lace then set in the previous question would contain what?  12. (1.00 pts) The list in the previous question would contain what?  A just 2. 4. 6. 8  C just 3. 5. 7  D just 3. 5. 7, 9  13. (1.00 pts) What is the average time complexity of prepending to a python list?  A just 3. 9 (1)  B just 3. 9 (1)  What is the average time complexity of prepending to a python list?  A just 3. 9 (1)  B just 3. 9 (1)  14. (1.00 pts) What is the average time complexity of retrieving an element from a dictionary?  A just 3. 9 (1)  B just 4. (1.00 pts) What is the average time complexity of retrieving an element from a dictionary?  A just 3. (1.00 pts) Suppose I had an ordered list of numbers of size n and I wanted to insert them into a Binary Search Tree? What would be the big-Q of this operation?	O C)repr_	_ should only be called by the system, whilestr can be called by users			
This would be an example of what python language concept?  A) List comprehension B) List generation C) List reduction D) For loop transformation  12. (1.00 pts) The list in the previous question would contain what?  A) [0, 2, 4, 6] B) [0, 2, 4, 6, 8] C) [1, 3, 5, 7] D) [1, 3, 5, 7, 9]  13. (1.00 pts) What is the average time complexity of prepending to a python list?  A) O(1) B) O(n) C) O(n)*2 D) O(log n)  What is the average time complexity of retrieving an element from a dictionary?  A) O(1) B) O(log n)  United to pure the complexity of retrieving an element from a dictionary?  A) O(1) B) O(log n)  Suppose I had an ordered list of numbers of size n and I wanted to insert them into a Binary Search Tree? What would be the big-O of this operation?	O D)str	returns a user friendly representation the object, whilerepr generally returns the raw object format			
This would be an example of what python language concept?  A) List generation C) List reduction D) For loop transformation  12. (1.00 pts) The list in the provious question would contain what?  A) (0, 2, 4, 6) B) (0, 2, 4, 6, 8) C) (1, 3, 5, 7, 9)  13. (1.00 pts) What is the average time complexity of prepending to a python list?  A) (01) B) (0) C) (0 m²2) D) (0 g n)  14. (1.00 pts) What is the average time complexity of retrieving an element from a dictionary?  A) (01) B) (0(s) n) C) (0(m²2) C) (0(n²2) D) (0(m²2) Suppose I had an ordered list of numbers of size n and I wanted to insert them into a Binary Search Tree? What would be the big-O of this operation?	11. (1.00 pts)	Suppose I had a Python Statement such as the following:			
A) List comprehension  B) List generation  C) List reduction  D) For loop transformation  12. (1.00 pts) The list in the previous question would contain what?  A) (0, 2, 4, 6)  B) (0, 2, 4, 6, 8)  C) (1, 3, 5, 7)  D) (1, 3, 5, 7, 9)  13. (1.00 pts) What is the average time complexity of prepending to a python list?  A) (11)  B) (0n)  C) (0n°2)  D) O(10g n)  14. (1.00 pts) What is the average time complexity of retrieving an element from a dictionary?  A) O(1)  B) O(0)  D) O(n°2)  D) O(n°2)  Suppose I had an ordered list of numbers of size n and I wanted to insert them into a Binary Search Tree? What would be the big-O of this operation?		[x for x in range(0, 10) if x % 2 == 0]			
<ul> <li>○ B) List generation</li> <li>○ C) List reduction</li> <li>○ D) For loop transformation</li> </ul> 12. (1.00 pts) The list in the previous question would contain what? <ul> <li>○ A) [0, 2, 4, 6]</li> <li>○ B) [0, 2, 4, 6, 8]</li> <li>○ C) [1, 3, 5, 7]</li> <li>○ D) [1, 3, 5, 7, 9]</li> </ul> 13. (1.00 pts) What is the average time complexity of prepending to a python list? <ul> <li>○ A) O(1)</li> <li>○ B) O(n)</li> <li>○ C) O(n*2)</li> <li>○ D) O(log n)</li> </ul> 14. (1.00 pts) What is the average time complexity of retrieving an element from a dictionary? <ul> <li>○ A) O(1)</li> <li>○ B) O(log n)</li> <li>○ C) O(n*2)</li> <li>○ D) O(n*2)</li> <li>○ D) O(n*2)</li> </ul> 15. (1.00 pts) Suppose I had an ordered list of numbers of size n and I wanted to insert them into a Binary Search Tree? What would be the big-O of this operation?		This would be an example of what python language concept?			
C C) List reduction  D) For loop transformation  12. (1.00 pts) The list in the previous question would contain what?  13. (1.00 pts) The list in the previous question would contain what?  13. (1.00 pts) What is the average time complexity of prepending to a python list?  13. (1.00 pts) What is the average time complexity of prepending to a python list?  14. (1.00 pts) What is the average time complexity of retrieving an element from a dictionary?  15. (1.00 pts) Suppose I had an ordered list of numbers of size n and I wanted to insert them into a Binary Search Tree? What would be the big-O of this operation?	O A) List con	nprehension			
Copy For loop transformation  12. (1.00 pts) The list in the previous question would contain what?  A) [0, 2, 4, 6] B) [0, 2, 4, 6, 8] C) [1, 3, 5, 7] D) [1, 3, 5, 7, 9]  13. (1.00 pts) What is the average time complexity of prepending to a python list?  A) [0(1) B) [0(1) D) [0(1)	O B) List gen	neration			
12. (1.00 pts) The list in the previous question would contain what?    A) [0, 2, 4, 6]	O C) List red	uction			
A) [0, 2, 4, 6] B) [0, 2, 4, 6, 8] C) [1, 3, 5, 7] D) [1, 3, 5, 7, 9]  13. (1.00 pts) What is the average time complexity of prepending to a python list?  A) O(1) B) O(n) C) O(n°2) D) O(log n)  14. (1.00 pts) What is the average time complexity of retrieving an element from a dictionary?  A) O(1) B) O(n) C) O(n°2) D) O(n°2) C) O(n) D) O(n°2) Suppose I had an ordered list of numbers of size n and I wanted to insert them into a Binary Search Tree? What would be the big-O of this operation?	O D) For loop	o transformation			
<ul> <li>B) [0, 2, 4, 6, 8]</li> <li>C) [1, 3, 5, 7]</li> <li>D) [1, 3, 5, 7, 9]</li> </ul> 13. (1.00 pts) What is the average time complexity of prepending to a python list?  A) O(1) B) O(n) C) O(n²2) D) O(log n)  14. (1.00 pts) What is the average time complexity of retrieving an element from a dictionary?  A) O(1) B) O(log n) C) O(n²2) D) O(n²2) Suppose I had an ordered list of numbers of size n and I wanted to insert them into a Binary Search Tree? What would be the big-O of this operation?	12. (1.00 pts)	The list in the previous question would contain what?			
C c) [1, 3, 5, 7] D) [1, 3, 5, 7, 9]  13. (1.00 pts) What is the average time complexity of prepending to a python list?  A) O(1) B) O(n) C) O(n'2) D) O(log n)  14. (1.00 pts) What is the average time complexity of retrieving an element from a dictionary?  A) O(1) B) O(1) B) O(10 D) O(n'2)  O) O(n'2)  Suppose I had an ordered list of numbers of size n and I wanted to insert them into a Binary Search Tree? What would be the big-O of this operation?	O A) [0, 2, 4,	6]			
O D) [1, 3, 5, 7, 9]  13. (1.00 pts) What is the average time complexity of prepending to a python list?  O A) O(1) O B) O(n^2) O D) O(log n)  14. (1.00 pts) What is the average time complexity of retrieving an element from a dictionary?  O A) O(1) O B) O(log n) O C) O(n) O D) O(n^2) Suppose I had an ordered list of numbers of size n and I wanted to insert them into a Binary Search Tree? What would be the big-O of this operation?	O B) [0, 2, 4,	6, 8]			
13. (1.00 pts) What is the average time complexity of prepending to a python list?  A) O(1) B) O(n) C) O(n^2) D) O(log n)  14. (1.00 pts) What is the average time complexity of retrieving an element from a dictionary?  A) O(1) B) O(log n) C) O(n) D) O(n^2)  15. (1.00 pts) Suppose I had an ordered list of numbers of size n and I wanted to insert them into a Binary Search Tree? What would be the big-O of this operation?	O C) [1, 3, 5,	7]			
A) O(1) B) O(n) C) O(n^2) D) O(log n)  14. (1.00 pts) What is the average time complexity of retrieving an element from a dictionary?  A) O(1) B) O(log n) C) O(n) D) O(n^2)  15. (1.00 pts) Suppose I had an ordered list of numbers of size n and I wanted to insert them into a Binary Search Tree? What would be the big-O of this operation?	O D) [1, 3, 5,	7, 9]			
<ul> <li>B) O(n)</li> <li>C) O(n^2)</li> <li>D) O(log n)</li> </ul> 14. (1.00 pts) What is the average time complexity of retrieving an element from a dictionary? <ul> <li>A) O(1)</li> <li>B) O(log n)</li> <li>C) O(n)</li> <li>D) O(n^2)</li> </ul> 15. (1.00 pts) Suppose I had an ordered list of numbers of size n and I wanted to insert them into a Binary Search Tree? What would be the big-O of this operation?	13. (1.00 pts)	What is the average time complexity of prepending to a python list?			
C) O(n^2) D) O(log n)  14. (1.00 pts) What is the average time complexity of retrieving an element from a dictionary?  A) O(1) B) O(log n) C) O(n) D) O(n^2)  15. (1.00 pts) Suppose I had an ordered list of numbers of size n and I wanted to insert them into a Binary Search Tree? What would be the big-O of this operation?	O A) O(1)				
14. (1.00 pts) What is the average time complexity of retrieving an element from a dictionary?  A) O(1) B) O(log n) C) O(n) D) O(n^2)  15. (1.00 pts) Suppose I had an ordered list of numbers of size n and I wanted to insert them into a Binary Search Tree? What would be the big-O of this operation?	O B) O(n)				
14. (1.00 pts) What is the average time complexity of retrieving an element from a dictionary?  O(1) O(1) O(1) O(1) O(1) O(1) O(1) O(1	O C) O(n^2)				
<ul> <li>A) O(1)</li> <li>B) O(log n)</li> <li>C) O(n)</li> <li>D) O(n^2)</li> <li>15. (1.00 pts) Suppose I had an ordered list of numbers of size n and I wanted to insert them into a Binary Search Tree? What would be the big-O of this operation?</li> </ul>	O D) O(log n)				
<ul> <li>B) O(log n)</li> <li>C) O(n)</li> <li>D) O(n^2)</li> <li>15. (1.00 pts) Suppose I had an ordered list of numbers of size n and I wanted to insert them into a Binary Search Tree? What would be the big-O of this operation?</li> </ul>	14. (1.00 pts)	What is the average time complexity of retrieving an element from a dictionary?			
C) O(n) D) O(n^2)  15. (1.00 pts) Suppose I had an ordered list of numbers of size n and I wanted to insert them into a Binary Search Tree? What would be the big-O of this operation?	O A) O(1)				
O D) O(n^2)  15. (1.00 pts) Suppose I had an ordered list of numbers of size <i>n</i> and I wanted to insert them into a Binary Search Tree? What would be the big-O of this operation?	O B) O(log n				
15. (1.00 pts) Suppose I had an ordered list of numbers of size <i>n</i> and I wanted to insert them into a Binary Search Tree? What would be the big-O of this operation?	O C) O(n)				
	O D) O(n^2)				
O A) Office n	15. (1.00 pts)	Suppose I had an ordered list of numbers of size <i>n</i> and I wanted to insert them into a Binary Search Tree? What would be the big-O of this operation?			
( ) At Chount	O A) O(log n	)			

O B) O(n log	○ B) O(n log n)					
	O C) O(n)					
O D) O(n^2)						
16. (1.00 pts)	Suppose I converted the Binary Search Tree in the previous problem into a Red-Black Tree. What would be big O of this operation?					
O A) O(1)						
O B) O(log n						
O C) O(n^2)						
O D) O(2^n)						
17. (1.00 pts)	Assume x is a list. What does the following line of python do?					
(						
	y = x[:]					
O A) Creates	s a deep copy of x and stores it in y					
O B) Creates	s a shallow copy of x and stores it in y					
O C) Nothing	, this is an example of a python noop					
O D) None o	f the above					
18. (1.00 pts)	If I had a program written in C and a program written in Python, both of which did the exact same thing, which program would have better performance and why?					
O A) C. beca	use it is a compiled language which runs much faster than Python					
	because it is an interpreted language and runs faster than C					
	ould have the exact same performance					
O D) C, beca	use C is a low-level language					
19. (1.00 pts)						
	1 = [] t = ()					
	<pre>s = set() f = frozenset()</pre>					
	d = {} i = 1					
	Check all the statements where the assertion fails.					
	Check all the statements where the assertion rails.					
(Mark <b>ALL</b> correct						
☐ B) assert t						
C) assert s						
☐ D) assert f	is frozenset()					
☐ E) assert o	t is {}					
☐ F) assert i	is 1					
20. (1.00 pts)	Given:					
<pre># numpy.shape(x) == (1, 10)</pre>						
	<pre># numpy.shape(M) == (20, 10) # k == 50</pre>					
What is the output of the following? Enter your answer in the form "(a, b)", where a and b are integers. If there is an error, type "ERROR".						
	<pre>print(numpy.shape(k * numpy.matmul(M,x.T))</pre>					

```
21. (1.00 pts) Given:
                 # numpy.shape(x) == (10, 1)
               What is the output of the following? Enter your answer in the form "(a, b)", where a and b are integers. If there is an error, type "ERROR".
                 print(numpy.shape(numpy.matmul(x,x.T))
22. (1.00 pts)
                 1 = []
                 t = ()
                s = set()
                 f = frozenset()
                d = {}
                 i = 1
                 s.add(t)
                 s.add(f)
                 s.add(d)
                 s.add(i)
               Are the above statements legal?
○ True ○ False
23. (1.00 pts)
                 #Child is a subclass of Parent
                 def f (num) :
                         if (num == 0)
                                  raise Parent(...)
                         if (num == 1)
                                  raise Child(...)
                 def g (num) :
                         try :
                                  a = f(num)
                         except Parent as e :
                                  print("hello")
                         except Child as e :
                                  print("world")
               What is the result of calling g(1)?
O A)
         hello
O B)
         world
O C)
         hello
         world
O D)
         <error>
24. (1.00 pts)
               Consider the following generator function myzip().
               myzip() takes in any iterable (list, set, etc.) and has the following behavior
                 Input:
                 a = [1,2,3]
                 b = [4,5,6]
                 c = [7,8,9]
                 Code:
                 x = myzip(a, b, c)
                 list(x)
                 output: [(1, 4, 7), (2, 5, 8), (3, 6, 9)]
```

Here is the implementation of myzip():

```
#assume iterables are all equal length
def myzip(*iterables):
   iterators = [MISSING for iterable in iterables]
   while iterators:
       result = []
       for it in iterators:
            temp = next(it)
            result.append(temp)
       yield tuple(result)
```

What goes in the place of MISSING?

```
25. (1.00 pts)
```

```
Input:
a = [1,2,3]
b = [4,5,6]
c = [7,8,9]

Code:
x = myzip(a, b, c)
list(x)

output: [(1, 4, 7), (2, 5, 8), (3, 6, 9)]
```

What is the additional output if we call list(x) again?

```
O A) [(1, 4, 7), (2, 5, 8), (3, 6, 9)]
```

- O B) []
- C) <syntax error>

26. (1.00 pts) Consider f() as an implementation of a factorial calculation. For example 4! = 24 and f(4) = 24.

The details of reduce() are described below:

```
def reduce(function, iterable, initializer=None):
    it = iter(iterable)
    if initializer is None:
        value = next(it)
    else:
        value = initializer
    for element in it:
        value = function(value, element)
    return value
```

```
def f(x):
    return reduce(lambda a,b : a*b, range(1, x + 1), 0)
```

There appears to be a bug in f(). f(4) returns \_\_\_\_ and to fix the bug, we must change the last argument in the function call to \_\_\_\_ or just remove it.

## 27. (1.00 pts)

In Python, you can pass functions as parameters and define functions inside functions. These nested functions cannot be accessed from the outside, but they allow us to build something called a decorator, which takes a function and modifies its behavior.

```
def mydecorator(f):
    def g():
        print("The result is:")
        f()
    return g

def myfunction():
    print("One")
myfunction = mydecorator(myfunction)
```

```
What is the result of a call to myfunction()?
O A)
         The result is:
O B)
         One
O C)
         <syntax error>
O D)
         <runtime error>
28. (1.00 pts)
                 def mydecorator(f):
                     def g():
                         print("The result is:")
                         f()
                     return g
               Let's redefine myfunction() to take in a parameter.
                 def myfunction(num):
                    print(num * num)
                myfunction = mydecorator(myfunction)
               Look carefully at mydecorator() and g() again. What is the result of a call to myfunction(3)?
O A)
         The result is:
O B)
O C)
         <syntax error>
O D)
         <runtime error>
29. (1.00 pts)
                 class Foo():
                    i = 1
                     def bar(self):
                         print (Foo.i)
                         print (self.i)
                 f = Foo() # Create an instance of Foo
               What is the output of f.bar()?
O A)
         1
O B)
O C)
O D)
         2
O E)
         <error>
```

30. (1.00 pts)

```
class Foo():
     i = 1
     def bar(self):
         print (Foo.i)
         print (self.i)
 f = Foo() # Create an instance of Foo
 Foo.i = 2
What is the output of f.bar() after we execute the above statement?
O A)
        1
O B)
O C)
        1
O D)
O E)
        <error>
31. (1.00 pts)
                class Foo():
                    i = 1
                    def bar(self):
                        print (Foo.i)
                        print (self.i)
                f = Foo() # Create an instance of Foo
                Foo.i = 2
                f.i = 1
              What is the output of f.bar() after we execute the above statements?
O A)
        1
        1
○ B)
        1
O C)
        1
O D)
        2
O E)
        <error>
32. (1.50 pts) Let A, and B be mutually exclusive events. What is P(A \cap B)?
○ A) P(B)
O B) P(A)P(B)
O C) P(A)
O D) 0
```

<b>33. (1.50 pts)</b> Let A, and B be independent events. What is P(A ∩ B)?
○ A) P(A)P(B)
○ B) P(A)
O C) P(B)
O D) 0
<b>34.</b> (1.50 pts) A binomial distribution can be expressed as which of the following?
A) The Product of Identical Poisson Distributions
O B) The Sum of Identical Hypergeometric Distributions
C) The Sum of Identical Poisson Distributions
O D) The Product of Identical Normal Distributions
<b>35. (1.50 pts)</b> Suppose the random variable X models a fair 6-sided dice roll. Determine its expected value E[X].
O A) 3
O B) 3.5
O C) 4
O D) None of the above
<b>36. (1.50 pts)</b> Suppose we knew the probability that a device was defective given the manufacturer. What could we use to find the probability that a certain manufacturer made the device, given that we have the probability that a device is defective?
○ A) Central Limit Theorem
B) Chebyshev inequality
O C) Jensen inequality
O D) Bayes' Theorem
<b>37.</b> (1.50 pts) Suppose the random variable X modes a fair six-sided dice roll. Determine its variance, var(X).
O A) 1.57
○ в) 2.92
O C) 3
O D) 3.5
38. (1.50 pts)  Farmer John wants to make sure that his cows are eating an adequate amount of food. Suppose the random variable F represents the amount of food a cow on Farmer John's farm gets. F is a Gaussian random variable with mean 10.5 and variance 5.3. Suppose Daisy, a cow on Farmer John's farm, eats 12.2 kg of food. Determine Daisy's percentile, rounded to the nearest whole percentage. You can look up a z-score table for this problem.
O A) 77th percentile
O B) 60th percentile
O C) 5th percentile
O D) 95th percentile
<b>39. (1.50 pts)</b> Consider a smooth probability distribution function P(x), defined for all real numbers x. The probability that x=3 is
O A) 0
O B) P(3)

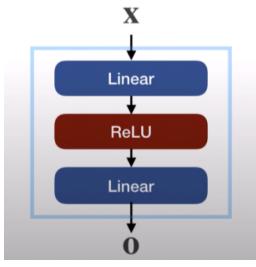
40. (1.50 pts) Which of the following is NOT TRUE about the mean and sample mean?
O A) For Gaussian distributions, the sample mean is an unbiased estimator of the mean.
O B) Given a distribution, the mean is a deterministic value whereas the sample mean is a random variable.
O C) The variance of the sample mean depends on the number of samples.
Op) The mean is equal to the sample mean for Poisson distributions.
41. (1.50 pts)  A statistician tests a random population sample and finds that 3% of them test COVID-positive. So she estimates the true COVID infection rate to be 3%. After doing more analysis, she finds a 95% confidence interval of [0%, 13%]. Which of the following is a valid interpretation of this statistical measure?
O A) 95% of people will test COVID-negative
O B) Between 0% and 13% of people have an 5% uncertainty of having COVID
O ) 95% of sample populations will have an infection rate between 0% and 13%
Op) The true infection rate is 95% likely to be between 3% and 16%
42. (4.00 pts)  The largest unit of computation that's structure is the same throughout neural network architectures is called a layer. Examples of layers include linear layers (fully connected layer, convolutional layer), activation functions (ReLU, Softmax, TanH, Leaky ReLU), pooling layers, loss functions, output transformations, etc.  X  Linear  Linear  Linear  What are linear layers? What type of functions can they represent? What do the innards of a linear layer look like given the input vector x? Why don't machine learning engineers put a linear layer directly after another linear layer?

## 43. (4.00 pts)

O C) P(1/3)

 $\bigcirc$  D) The area under the curve between x=0 and x=3

The largest unit of computation that's structure is the same throughout neural network architectures is called a layer. Examples of layers include linear layers (fully connected layer, convolutional layer), activation functions (ReLU, Softmax, TanH, Leaky ReLU), pooling layers, loss functions, output transformations, etc.



What are activation functions and why do we use them? Which of the following binary functions requires an activation function for a neural net to "learn" it: AND, OR, XOR, NOT, NAND, NOR?

44. (3.00 pts)	When is it a good idea to use a clustering algorithm (i.e. k-means, DBSCAN, etc) vs a deep learning model?		
45. (4.00 pts)  Principal component analysis is a linear transformation that decomposes a high-dimensional vector space into the directions that best explain the variance in a data set. Explain how this relates to dimensionality reduction (one of the most common applications of PCA).			

Thank you for competing in Data Science! Good luck with your other events!

© 2020 - powered by Scilympiad (https://scilympiad.com)

Support (/ut-invite/Support) | Contact (/ut-invite/Home/Contact)