CS 220 Spring 2020
Instructor: Tyler Caraza-Harter

Midterm — 20%

(Last) Surname: _____ (First) Given name: _____

NetID (email): _____ @wisc.edu

Fill in these fields (l_____, ____) scantron form (use #2 pencil):

1. LAST NAME (surname) and FIRST NAME (given name), fill in bubbles
2. IDENTIFICATION NUMBER is your Campus ID number, fill in bubbles
3. Under *A* of SPECIAL CODES, write your lecture number, fill in bubbles:
    1 - MWF 9:55am (Tyler morning)
    2 - MWF 2:25pm (Tyler afternoon)
4. Under *B* of SPECIAL CODES, tell us about the nearest person (if any) to your left
    0 - no person to the left in your row
    1 - somebody you **do not** know is there
    2 - somebody you **do** know is there
5. Under *C* of SPECIAL CODES, do the same as *B* but for the person to your right
6. Leave special codes *DE* blank
7. Under *F* of SPECIAL CODES, write **8** and fill in bubble **8**

....

Make sure you fill all the special codes above accurately in order to get graded.

....

You have 2 hours to take the exam. Use a #2 pencil to mark all answers. When you're done, please hand in these sheets in addition to your filled-in scantron.

You may not sit adjacent to your friends or other people you know in the class (having only one empty seat is considered "adjacent"). You may only reference your notesheet. You may not use books, your neighbors, calculators, or other electronic devices on this exam. Please place your student ID face up on your desk. Turn off and put away portable electronics now.

We won't always include imports in examples, but assume it has been done in a standard way (for example, assume `from pandas import DataFrame, Series` before each code example). Sometimes there will be multiple right answers – choose the best one. For example, saying an order of growth is in O(N) is more informative (and therefore better) than saying it is in O(N**2), assuming both are true.

程序代写代做 CS编程辅导

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

https://tutorcs.com

Trees

root: 6
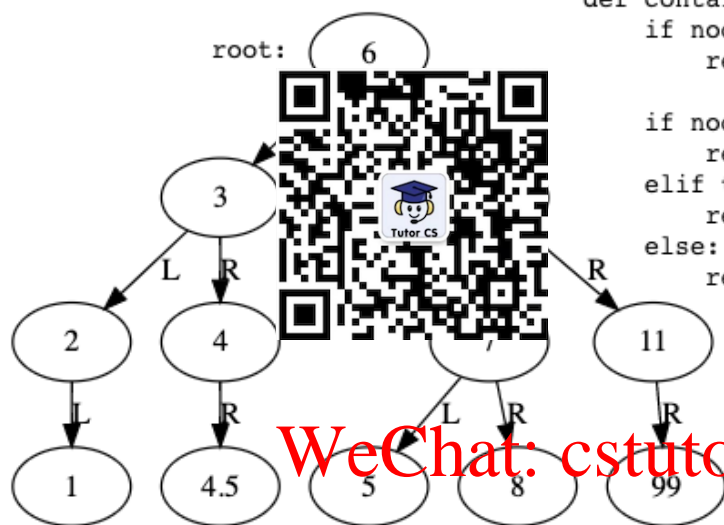
```
def contains(node, target):
    if node == None:
        return False

    if node.val == target:
        return True
    elif target < node.val:
        return contains(node.left, target)
    else:
        return contains(node.right, target)
```

Tree structure (from diagram):
- root: 6
- 3 (left child of 6)
  - 2 (L of 3)
    - 1 (L of 2)
  - 4 (R of 3)
    - 4.5 (R of 4)
- 7 (7 is under root, with children)
  - 5 (L)
  - 8 (R)
- 11 (R of 6)
  - 99 (R of 11)

"R" means "right child"
"L" means "left child"

1. Which leaf should be removed to make the above binary tree a BST?

   A. 1   B. 4.5   **C. 5**   D. 8   E. 99

2. What is `root.left.right.left`?

   A. None   B. node 1   C. node 2   D. node 4   E. node 4.5

3. If somebody calls `contains(root, 12)`, how many times will `contains` be invoked? Assume `root` refers to node 6.

   A. 1   B. 2   **C. 4**   D. 6   E. 12

4. What is/are the *base case(s)* for `contains`?

   A. when `node == None`
   B. when `node.val == target`
   C. all of the above
   D. none of the above

5. (T/F) When inserting nodes to a BST, inserting in random order will typically produce a **more balanced** tree than inserting in ascending order.

   A. True   B. False

Complexity

Assume a correct and efficient merge_sort implementation (like the one in lab 3). Let N be len(nums).

```python
def cumulative_v1(nums):
    nums = merge_sort(nums)

    results = []
    for i in range(len(nums)):
        total = 0
        for j in range(0, i+1):
            total += nums[j]
        results.append(total)
    return results


def cumulative_v2(presorted_nums):
    results = deque()
    total = 0
    for x in presorted_nums:
        total += x
        results.appendleft(total)
    return results
```

6. (T/F) replacing `results.append(total)` with `results.insert(0, total)` in the first function would make the code more efficient.

   A. True   **B. False**

7. What is the time complexity of `cumulative_v1`?

   A. O(1)   B. O(log N)   C. O(N)   D. O(N log N)   **E. O(N**2)**

8. What is the time complexity of `cumulative_v2`?

   A. O(1)   B. O(log N)   **C. O(N)**   D. O(N log N)   E. O(N**2)

9. Which orders of growth are equivalent to O(2 * N**2)?

   A. O(2 * N**3)   **B. O(3 * N**2)**   C. all of the above   D. none of the above

10. A loop that iterates 100 times, each time printing the **length** of a list that contains **N** elements (and doing nothing else) has what complexity?

    A. O(1)   B. O(log N)   C. O(N)   D. O(N log N)   E. O(N**2)

Web

```python
from flask import Flask, request
app = Flask(__name__)

@app.route("/a.ht
def b():
    with open("a.        f:
        return f.

@app.route("/b.ht
def a():
    q = "<html><body>{}</body></html>".format("B" + "e"*10)
    return q

@app.route("/c.html")
def c():
    mult = int(request.args.get("mult", "1"))
    return "ha" * mult

if __name__ == "__main__":
    app.run(host="127.0.0.1")
```

11. Which function runs when "http://localhost/a.html" is visited? Careful, trick question!

    A. a()   **B. b()**   C. c()

12. Which page is static?

    A. a.html   B. b.html   C. c.html

13. Which page uses a query string?

    A. a.html   B. b.html   **C.** c.html

14. `@app.route("/a.html")` is an example of a(n):

    A. attributer   **B. decorator**   C. firewall   D. IP address   E. POST request

15. (T/F) The following will find at least on match:

    `re.findall(r"^\d\d", "K9")`

    A. True   **B. False**

Graphs

Assume the following search method is in a Node class. Assume the method is correct, with self.name referring [obscured] "1", "2", etc.). Assume self.[obscured] graph object (not shown) and sel[obscured].

```python
def search(self[obscured]
    if self.nam[obscured] visited:
        return [obscured]
    self.graph.[obscured]name)

    if self.name == dst:
        return (self.name, )
    for child in self.child[obscured]:
        childpath = child.search(dst)
        if childpath:
            return (self.name, ) + childpath
    return None
```
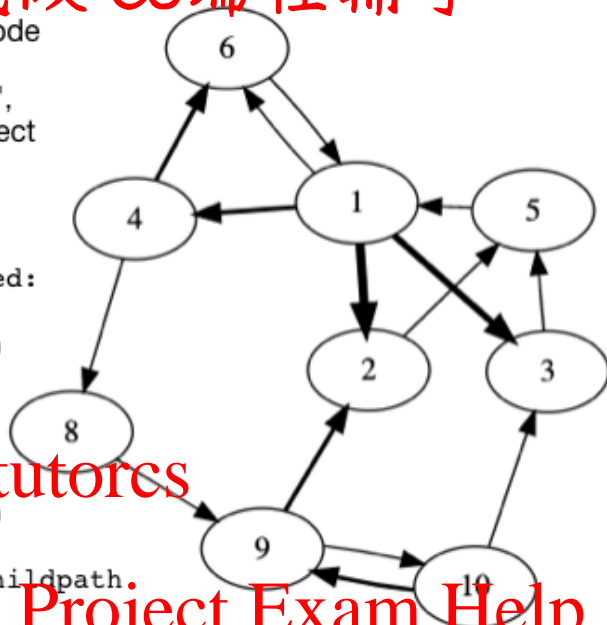
16. (T/F) The search method is an example of depth-first search.

    A. True   B. False

17. What is the graph an example of? Choose the **most informative** answer.

    A. directed graph   B. DAG   C. tree   D. BST

18. What is the shortest path from node 9 to node 1?

    A. 9,1   B. 9,2,1   **C. 9,2,5,1**   D. 9,10,3,5,1

19. If `node4` refers to the node with name "4", which of the following nodes will be checked **last** by `node4.search("11")`? By "checked", we mean this part: `self.name == dst`. Hint: node "6" will be visited before node "8".

    A. 2   B. 3   C. 5   D. 9   **E. 10**

20. If `search` were modified, what kind of coding mistake could cause a stack overflow?

    A. not adding nodes to `visited`
    B. iterating over node children in reverse order
    C. deleting the line that makes the recursive call to `search`
    D. changing all the returns to return `None`

Object Oriented Programming

```python
class Vehicle:
    def honk(self, mult=1):
        print("be...

class Bus(Vehicle...
    def __init__(
        self.capa...
        self.fill...

    def board(self, passengers):
        passengers = min(passengers, self.capacity-self.filled)
        self.filled += passengers
        return passengers

class Car:
    pass

v = Vehicle()
b1 = Bus(32)
b2 = Bus(32)
c = Car()
b1.board(30)
result = b2.board(4)
```

21. Which call will fail?

    A. b1.honk()   B. b2.honk(3)   C. c.honk(3)   D. v.honk(3)

22. What will `result` be?

    A. 0   B. 2   **C. 4**   D. 32   E. 64

23. What special method does `print` prefer to use, if available?

    A. __init__   B. __repr__   **C. __str__**
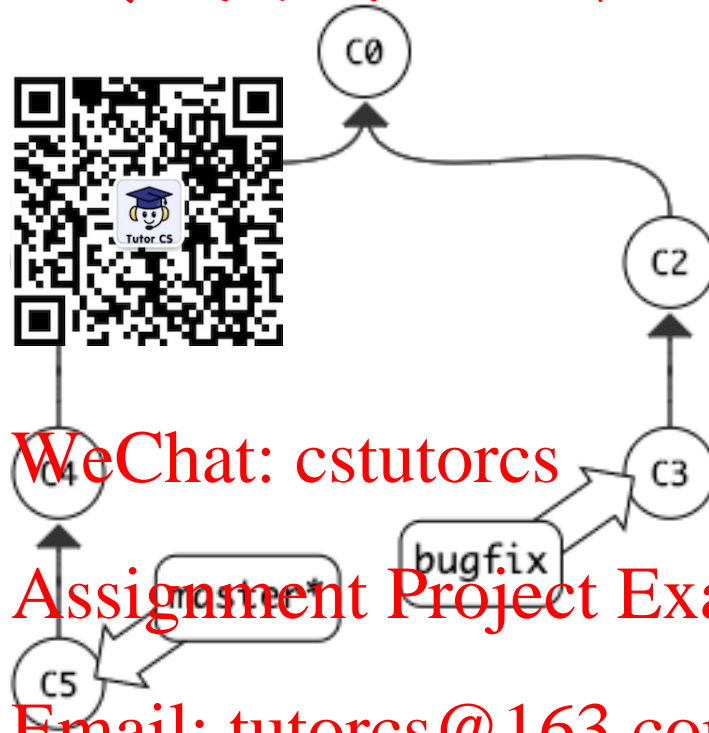
24. In this code, `self` is the name of a(n):

    A. attribute   B. class   C. constructor   **D. receiver parameter**

25. (T/F) Calling `b2.capacity()` will return 32.

    A. True   **B. False**

Reproducibility

26. (T/F) The current state is in "headless" mode.

    A. True   **B. False**

27. Which command will create a new commit with changes from both C5 and C3?

    A. add   B. commit   **C. merge**   D. branch   E. checkout

28. What does `time.time()` return?

    A. `datetime` object representing current time
    B. milliseconds to run entire program
    C. milliseconds since 1950
    D. seconds since 1970
    E. seconds to run the prior piece of code

29. What type does `subprocess.check_output(...)` return by default?

    A. bool   **B. bytes**   C. list   D. process

30. What is a running program is called?

    A. binary   B. executable   C. jogger   D. port   **E. process**

程序代写代做 CS编程辅导

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

https://tutorcs.com

(Blank Page: you may tear this off for scratch work, but hand it in with your exam)

程序代写代做 CS编程辅导

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

https://tutorcs.com