106 Midterm 2 Review

These questions are designed to help you think about course material, not to show you actual exam questions (though some are). Answers will be short answer, short essay, or coding.

Research shows that the best way to study for an exam is with other people. In group study, the people who start knowing more learn more (think about why), so don't think that it is only valuable to study with people who know more than you do.

- 1. Everything from midterm one, plus
 - inserting an element into an ordered list
 - insertion sort
 - merge (either version)
 - mergesort (recursive version from class only)
 - binary search
 - complexity: O(log n), O(nlog n), O(n), O(n**2), O(2**n)
 - nested loops
 - recursion
 - slicing notation
 - append, remove, insert
 - integer division, remainder (%)
 - dictionaries
 - zip
 - timing (not the specific fcns, but strategy)
 - tuples
 - what is False?
- 2. A typical question about a sorting algorithm will present you with code and require you to fill in blanks, or show the result of print calls in the function.
- 3. Show the values of all variables of the insertion sort function after each iteration when called on this list: [5, 1, 3, 4, 2, 7, 8]
- 4. Given the list [5, 1, 3, 4, 2, 7, 8] show slicing notation to a) remove the 1; b) replace both the 4 and 2 with a single 9; c) add the list [3, 3] to the end.

```
5. def splice(alist,blist):
    indexA = 0
    indexB = 0
    result = []
    while indexA < len(alist) and indexB < len(blist):
        print(result)
        if alist[indexA] < blist[indexB]:
            result = result + [alist[indexA]]</pre>
```

```
indexA += 1
else:
    result = result + [blist[indexB]]
    indexB += 1

if not indexA:
    result = result + alist[indexA:]
elif not indexB:
    result = result + blist[indexB:]
return result
```

6. (15 pts) Consider the code above. Do not change the code.

This function is NOT the same as the one from class, so follow the code *very* carefully. Note the two print calls (one above in the function, one below with the call). Trace the code and show what would print. You may not need all lines.

- 7. Explain why insertion sort is $O(n^2)$ (or why merge, or why mergesort, etc.)
- 8. Consider the function max(alist) that finds the largest number in alist. What is its complexity? Explain.
- 9. Create a dictionary "school" that associates the names of three students with their id numbers. Show how to get Mary's id out of the dictionary. Show how to add a new student.

- 10. Assume a 2-D nested list of integers of any size, e.g. [[1,2,3],[4,5,6],[7,6,5]]. Write a function to return the x,y indices of the first 6 in the list. (OR: sum the elements; sum the odd elements; find the max; count the 7s; replace any 5 with 15; etc.)
- 11. (18 pts) Assume you are *given* a function that returns the distance between any two domesticated animals.

Write a function that takes a chicken and a list of cows and returns the closest cow to the chicken.

- 12. (6 pts) Assume you have a list of 4 billion cows, and you want to sort them. Using mergesort, about how many comparisons will this take? Show your work to receive credit.
- 13. (6 pts) Xenon has to sort 1 million items. Xenon forgot to study mergesort, quicksort, heapsort, Shell sort, or bucket sort, and so is using insertion sort. How long will it take on a 8 megahertz processor? Your answer should be in sensible units. Show your work to receive credit.
- 14. Given a partial solution and a maze, show what expand_partial_solution would return.
- 15. Given a list of partial solutions, which one would be expanded next? Why?
- 16. Xenon proposes to substitute the rating of a path with a metric based solely on the rating of a square. Why will this produce suboptimal results?
- 17. Suppose the definition of a square was changed from a dictionary to a class. What parts of your project would have to change, assuming you carefully followed instructions? Show the changes you would make. What software engineering concept was illustrated by this feature of the project?
- 18. Write a function that takes one parameter, a list, uses no helper functions or mutation, and reverses the list. Hint: use an index that counts down from the length of the list OR add elements to the front of a result list.¹
- 19. Correctly write your your TAs' first and last names.
- 20. What is truth? What is truth in Python? Which operators return booleans? What values can be interpreted as booleans?
- 21. What does this error msg mean, and how might you fix it?

TypeError: unsupported operand type(s) for +: 'int' and 'NoneType'

- 22. What advantages are typically associated with short code? When would short code be bad?
- 23. List three ways, discussed in class, to improve your own learning during class.
- 24. Any question from the topics of midterm 1! Also, all labs to date.
- 25. What are the features of an effective recursive function?

¹Try this (not for exam): write it so that it also reverses all nested lists, e.g. [1,[2,3][4,[5,6],7]] =[[7,[6,5],4],[3,2],1]

- 26. When we write mergesort recursively, it goes on the stack more than once. What is the maximum number of times it can appear on the stack at once?
- 27. Given partial code for merge, mergesort or insertion sort, fill in the blanks.