Sample Midterm

October 20, 2019

- The first midterm is Saturday, October 26 from 10:00 AM to 12:00 PM.
- You will not be able to bring any books or electronic devices.
- Bring your photo ID.
- This is not a practice exam. An actual exam will have possibly quite different problems and will be harder.
- It is strongly recommended that you go over all materials, e.g., lecture notes, homeworks, quizzes, and jupyter notebooks.

Problem 1. What is the output of the following Python program?

```
sum = 0
n = 10
while n > 0:
  sum = sum + n
  n = n - 2
print(sum)
```

Problem 2. Use list comprehension to make all strings in the following list lowercase:

```
L = ['Seoul', 'NEW␣YORK', 'Paris', 'Hello␣World', 'SNU']
```

Problem 3. What is the output of the following program?

```
s = 'hello!'
print(s * s.count('l'))
```

Problem 4. What are the contents of l2 after the following code is executed?

```
L1 = ['Monday', 'Tuesday', 'Wednesday', 'Thursday']
L2 = [(L1[i], L1[i+1]) for i in range(len(L1)-1)]
```

Problem 5. What is the value of the expression `D['one']` after the following code is executed?

```
words = ['one', 'two', 'One', 'Three', 'ONE', 'Two', 'three']
D = {}
for i in range(len(words)):
  w = words[i].lower()
  if w in dct:
    D[w][0] += 1
    D[w][1] = i
  else:
    D[w] = [1, i]
```

Problem 6. What is the output of the following Python program?

```
def fun(a, b=5, c=10):
  return a+b+c

print(fun(2,3))
```

Problem 7. For each of the functions $f(N)$ given below, indicate the tightest bound possible. You must choose your answer from the following:

$$O(1), O(N), O(N^2), O(N^3), O(N^4), O(N^5), O(\log N), O(N \log N), O(N^2 \log N), O(N^3 \log N), O(2^N)$$

(a) $f(N) = N(N^2 \log N^2 + N)$

(b) $f(N) = (10N + N^2)^2$

(c) $f(N) = N \log 100000$

Problem 8. Draw the recursion trace for the computation of `reverse(’abcdefg’, 0, 6)` as implemented in the following code fragment:

```
def reverse(S, start, stop):
  if start < stop - 1:
    temp = S[start]
    S[start] = S[stop-1]
    S[stop-1] = temp
    reverse(S, start+1, stop-1)
```

Problem 9. Describe the worst case running time of the following function in Big-Oh notation in terms of the variable `n`.

```
def fun(n):
  if n < 1:
    return n
  elif n < 100:
    return fun(n - 1)
  else:
    return fun(n // 2)
```

Problem 10. We want to implement a recursive binary search algorithm for a list which assumes the underlying data is sorted in descending order, e.g., `[10, 7, 5, 1]`. Please complete the commented line.

```
def binary search_descending(data, target, low, high):
  if low > high:
    return False
  else:
    mid = (low + high) // 2
    if target == data[mid]:
      return True
    elif target < data[mid]:
      # please complete this line
    else:
      # please complete this line
```

Problem 11. What is the output of the following Python script?

```
class Person():
  def __init__(self, name, idnumber):
    self.name = name
    self.idnumber = idnumber

  def display(self):
    print(self.name)
    print(self.idnumber)

class Employee(Person):
  def __init__(self, name, idnumber, salary):
    Person.__init__(self, name, idnumber)
    self.salary = salary
  def display(self):
    super().display()
    print(self.salary)

Sam = Person('Sam', 4439)
Sam.display()

Paul = Employee('Paul', 1234, 120000)
Paul.display()
```

Problem 12. Suppose that L is a Python list of length `n`. Categorize the worst-case execution time of the below expressions as either

$$O(1), O(\log n), O(n), O(n^2).$$

(a) `L.insert(2,10)`

(b) `L[-1]`

(c) `3 in L`

(d) `len(L)`

Problem 13. What is the output of the following Python program?

```
def fun(x, y):
  try:
    ratio = x / y
  except ZeroDivisionError:
    ratio = 'inf'
  return ratio

print(fun(3,2))
print(fun(2,0))
```

Problem 14. What values are returned during the following sequence of queue operations, if executed on an initially empty queue?
`enqueue(5)`, `enqueue(3)`, `dequeue()`, `enqueue(2)`, `enqueue(8)`, `dequeue()`, `dequeue()`, `enqueue(9)`, `enqueue(1)`, `dequeue()`, `enqueue(7)`, `enqueue(6)`, `dequeue()`, `dequeue()`, `enqueue(4)`, `dequeue()`, `dequeue()`.