Middlebury

CSCI 146: Intensive Introduction to Computing

Fall 2025

Final Review

Question 1

You are given four programs, each uses one of four different implementation for searching a sorted array: iterative linear search, recursive linear search, iterative binary search, recursive binary search. Unfortunately you don't know which program uses which approach. Which program uses which approach?

1. Search called five times with lists of length 433, 432, 431, 430, 429. recursive linear search
2. Search called once with list of length 1000.
3. Search called once with list of length 1000.
4. Search called five timeswith lists of length 16, 8, 4, 2, 1 recursive binary search

iterative linear or binary search

how to determine which is which?

experiment?

linear search O(n) → probably double runtime
double input list size

binary search O(logn) → minimal effect on runtime

Question 2

What is the Big-O worst-case time complexity of the following Python code? Assume that `list1` and `list2` are lists of the same length. n

indent missing

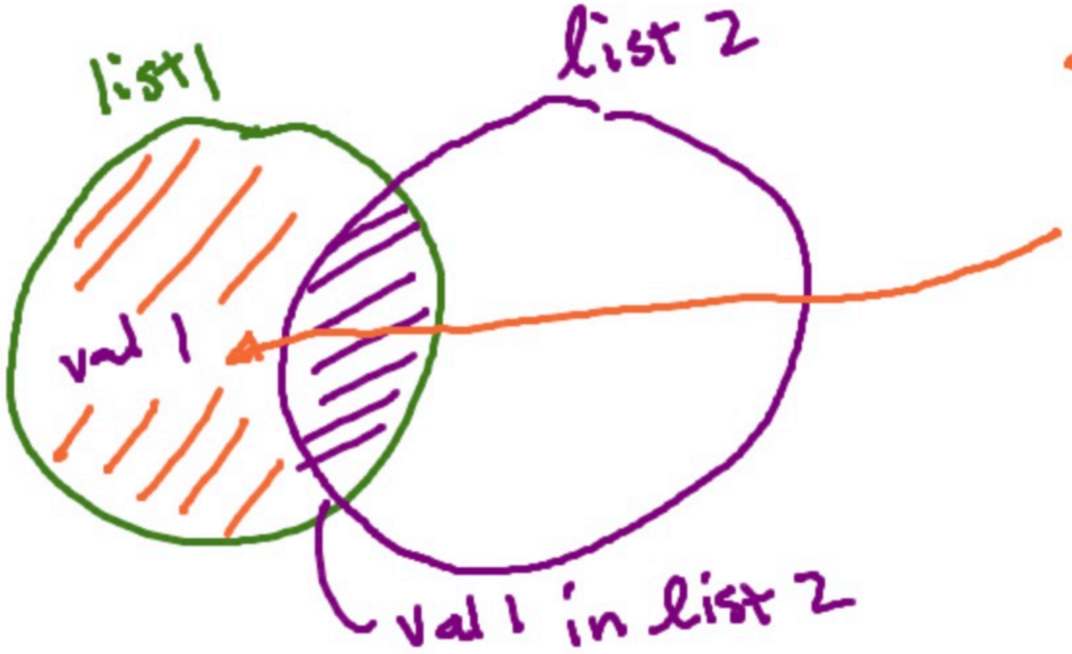
```
def difference(list1, list2):
result = []
for val1 in list1:
    if val1 not in list2:
        result.append(val1)
return result
```

→ n times

→ big O of "in" for list O(n)

how many iterations

O(n²)



S1 = set(list1)

S2 = set(list2)

S1 - S2

Question 3

What decimal numbers are represented by the following binary numbers?

$2^3 \, 2^2 \, 2^1 \, 2^0$

1. 1101 $8+4+1 = 13$
2. 111 $4+2+1 = 7$
3. 10010 + 11011

$$\begin{array}{rl} 1 \, 0 \, \overset{1}{0} \, 1 \, 0 & \rightarrow 18 \\ + \quad 1 \, 1 \, 0 \, 1 \, 1 & \rightarrow 27 \\ \hline 1 \, 0 \, 1 \, 1 \, 0 \, 1 & \end{array}$$

45

32 8 4 1 = 45 ✓

Question 4

Translate the following function using NumPy to just use Python built-ins assuming `a_list` is a list of floats (instead of a NumPy vector) and `lower` is a single (scalar) float:

→ scalar

```python
def sum_above(a_list, lower):
    return np.sum(a_list[a_list > lower])
```

for ?

if ?

True/False depending on whether an elem of a_list is > lower

```python
def sum_above(a_list, lower):
    result = 0
    for x in a_list:
        if x > lower:
            result += x
    return result
```

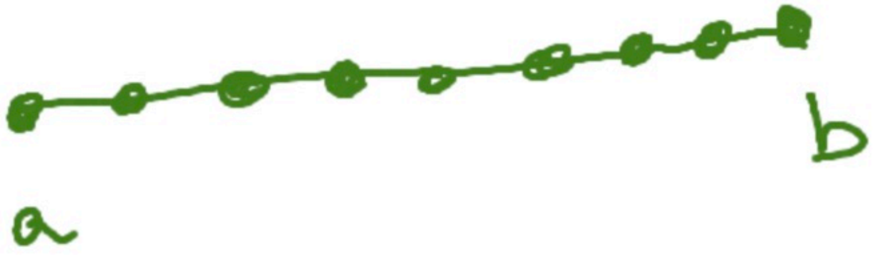
Question 5

Translate the following code to use NumPy so that it doesn't use any plain Python `for`-loops or `if` statements. Recall the `linspace` function can be used to create evenly spaced points on an interval. The goal is to calculate `n`.

```
lst = []
for i in range(101):
    lst.append(i)

n = 0
for x in lst:
    if x ** 2 < 30 * x:
       n += 1
```

```
import numpy as np
x = np.linspace(0, 100, 101)
n = np.sum( x**2 < 30*x)
```

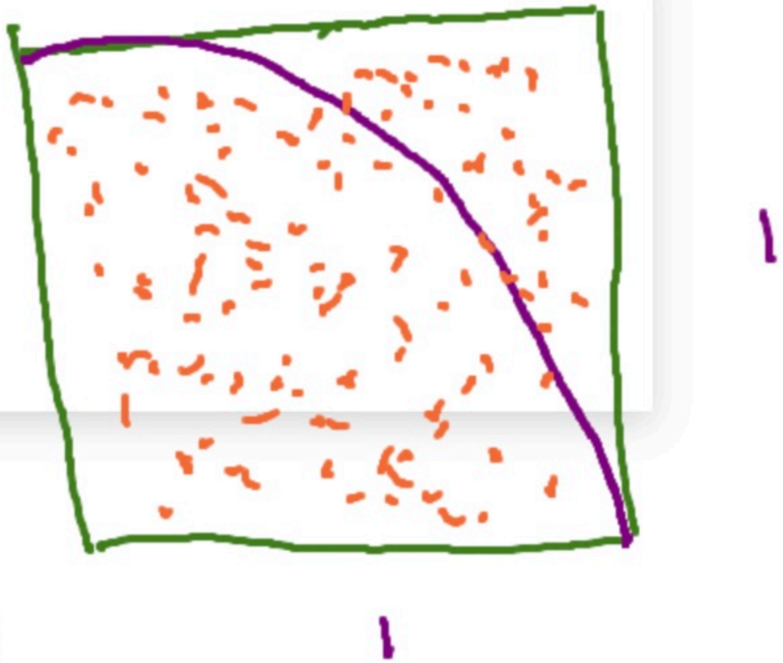


Question 6

The following code estimates π by creating n random points in a unit square and then determining the number of points (m) which fall into a quarter circle of radius 1. The fraction m/n is approximately equal to the fraction of the area of this quarter-circle ($\pi/4$) to the area of the square. Therefore $\pi \approx 4(m/n)$. Translate the code below to using only NumPy (no `for`-loops or `if`-statements) and note that `np.random.sample(num)` returns an array of `num` random numbers where each number r is $0 \leq r < 1$.

```
import random
n = 100000
m = 0
for _ in range(n):
  x = random.uniform(0, 1)
  y = random.uniform(0, 1)
  if x ** 2 + y ** 2 <= 1:
      m += 1
print("Pi is about " + str(4 * m / n))
```

2 arrays for x, y coordinates



memory needed to store x,y

```
x = np.random.sample(n)
y = np.random.sample(n)
m = np.sum(x**2 + y**2 <= 1)
pi = 4*m/n
```

Question 7

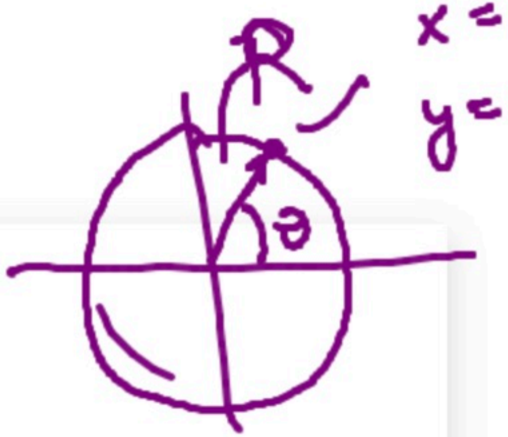
Draw the result of the following `numpy` and `matplotlib` code.

```python
import math
import numpy as np
import matplotlib.pyplot as plt

theta = np.linspace(0, 2 * math.pi)
x = 0.25 + 0.5 * np.cos(theta)
y = 0.25 + 0.5 * np.sin(theta)
plt.plot(x, y)
plt.show()
```

np.pi

R

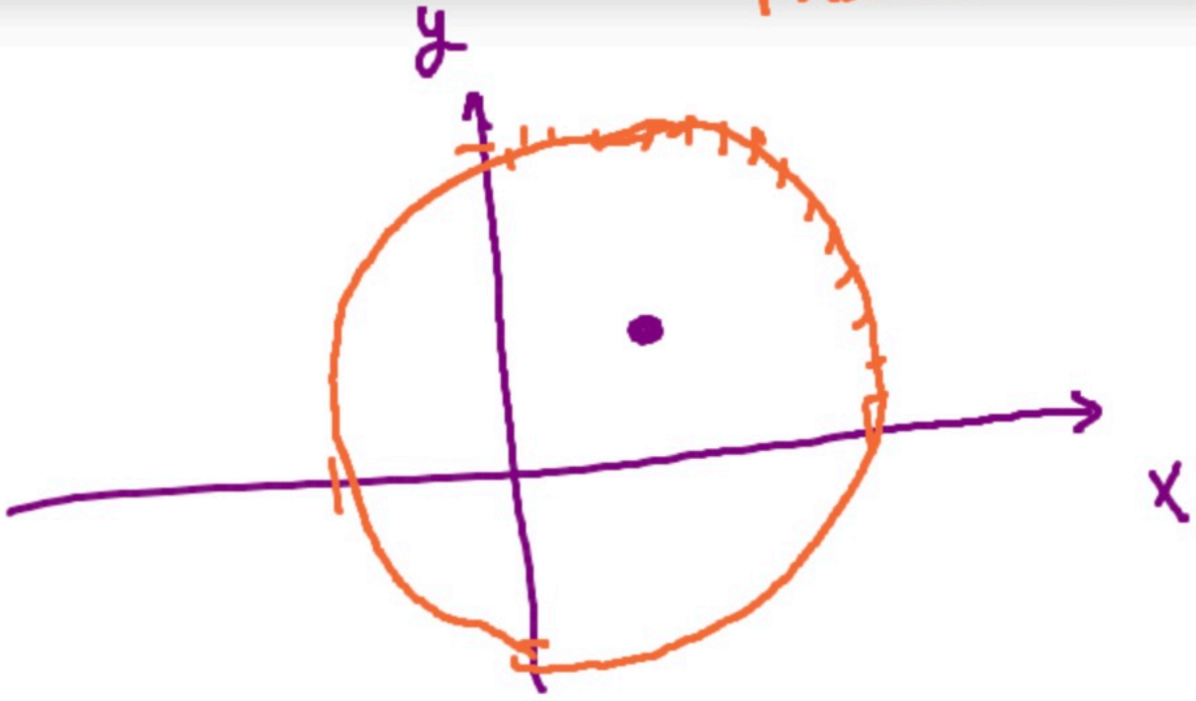


$x = R\cos\theta$

$y = R\sin\theta$

center is at (0.25, 0.25)

radius = 0.5



Final note about using Python after this class.

Using `middpy`:

- Option 1: You can continue working in your `cs146` folder.
- Option 2: You can create a new folder and copy the `.vscode` folder to this folder. Then click `Python Setup` again. Or install packages from the terminal using `pip` (e.g. `python -m pip install numpy`). See the "packages" list in `.vscode/settings.json`.

Not using `middpy`:

- Option 1: Use the play button at the top-right of VS Code (in ANY folder, as long as Python is installed and a Python interpreter is selected in VS Code). But note that this won't be "interactive" the way we have been using it.
- Option 2: From the terminal, call `python3 myscript.py` (non-interactive) or `python3 -i myscript.py` (interactive). Omit the `3` on Windows (i.e. just write `python` instead of `python3`).

Question 8

The word COMPUTER means...

with | together