

# Python in 4 sittings

## Problem Set 4

4th July, 2021

© [pi4py.netlify.app](https://pi4py.netlify.app) (<https://pi4py.netlify.app>)

feel free to contact us;

[arabindo@protonmail.com](mailto:arabindo@protonmail.com) (<mailto:arabindo@protonmail.com>),

[kaustavbasu97@gmail.com](mailto:kaustavbasu97@gmail.com) (<mailto:kaustavbasu97@gmail.com>).

**This problem set consists of two part - 1. Animation based, 2. Algorithmic problem**

## Simple Animation Based

### Problem 1:

**Take a simple sinusoidal function, which is varying in time. And animate that function using FuncAnimation.**

Say,

$$f(x, t) = \sin(5x - 2t)$$

In [ ]:

### Problem 2(a):

**Say you have  $f(x, t) = \frac{2}{l} \sin(\frac{n\pi x}{l} \cos(\omega_n t))$  and  $g(x, t) = \frac{2}{l} \sin(\frac{n\pi x}{l} \sin(\omega_n t))$ . Animate these function.**

Choose  $l=5$  and set the  $x$  limit from 0 to 5. Choose  $\omega_n = 1.77$

**Also show the behaviour of  $(f(x))^2 + (g(x))^2$**

In [ ]:

## Problem 3(b):

Now, suppose,

$$f(x, t) = \sqrt{\frac{30}{a}} \left(\frac{2}{\pi}\right)^3 \sum_{n=1,3,5,\dots} \frac{1}{n^3} \cos(n\pi x/a) \cos\left(\frac{n^2\pi^2 t}{2a^2}\right)$$

$$g(x, t) = \sqrt{\frac{30}{a}} \left(\frac{2}{\pi}\right)^3 \sum_{n=1,3,5,\dots} \frac{1}{n^3} \sin(n\pi x/a) \cos\left(\frac{n^2\pi^2 t}{2a^2}\right)$$

$$h(x, t) = (f(x, t))^2 + (g(x, t))^2$$

Use a for loop to get the sum within the animate function and plot/animate this function.

Try for different number of terms. Say, first 10 term then do it for first 20 terms and so on!

In [ ]:

## Algorithmic Problem

Link to the one of the possible solution is at the end of this section.

## Statement

Suppose, you have a very durable glass ball. You want to test the hardness by dropping it from different floors of a building. If a ball falls from  $n-1$ th floor and stays intact, then, we need to climb up and drop it again. Now, if we drop it from the  $n$ th floor and it breaks apart, we will define  $n$  as the quality factor of that glass ball.

## Problem 1(a):

First store a random integer (take a range of 1 to 100), which will be the quality factor of the ball.

Now design a program which will search for that  $n$  and will give you the correct output and number of iteration it took to found the correct value. For a naive solution It will take  $n$  number of iterations to find the correct value.

In [ ]:

## Problem 1(b):

Now optimize your code in such a way that the program can find  $n$  by using  $k$  iterations where  $k$  is very less than  $n$ . ( $n > k \geq \log(n)/\log(2)$ ) p.s; We are rich now., so, instead of using a single ball, now we can afford to break  $k$  balls (At MOST).

In [ ]:

## Problem 1(c):

Ohkay! The number of balls you have wasted in the last round have made us bankrupt and we only have two balls(No Pun Intended) which were left behind by the authorities. Solve the given problem using  $m$  number of iterations such that,  $m < n/2$

In [ ]:

## Problem 1(d)

Solve the given problem using no balls.

In [ ]:

## Problem 1(e):

Find an algorithm which can help us to clean the shattered glassballs from the  $m \times n$  floor. It should be done using atmost  $\max(m,n)$  steps.

In [ ]:

## Problem 1(f):

Give us 1 million USD. We will pray for you.

**Tata.**

In [ ]:

You may look at here for the solution:

<https://commandantekaustav.github.io/algo/prob/algorithmic%20problems.html>

[.https://commandantekaustav.github.io/algo/prob/algorithmic%20problems.html](https://commandantekaustav.github.io/algo/prob/algorithmic%20problems.html)) (in C++)