***The Sample Test***

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_****Python Asignment:***

*#Write a function to implement* ***binary search*** *which will take a sorted list and a number. It returns the index where number is present in the list else nil*

*#Example: def binarySearch(sorted\_list, number)*

*#Create a* ***django models*** *with User, Address and Remark tables. User can have mutiple addresses and one Remark*

*#Then add 3 users with 3 addresses each and remark*

*# feel free to choose fields inside model*

*#Implement a* ***max heap*** *using python class. Heap should contain an user object and it should heapify based on \_id*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*This application is built for sample test.*

***Status:*** *In progress....*

***Ubuntu Release:*** *Ubuntu 20.04.1 LTS*

***Utilities List:***

* ***Binary Search***
* ***Django Models***
* ***Max Heap Using Heapq***

***Binary Search***

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_def binarySearch(sorted\_list, number):*

*"""*

*Input:*

*sorted\_list: a sorted list*

*number: It is given item of information that to be serch in the sorted list*

*OutPut:*

*the index where number is present in the list else nil*

*Note: As per a given example iteration method is expect for the solution.*

*"""*

*beg = 0*

*end = len(List) - 1*

*mid = int( (beg + end)/2 )*

*while beg <= end and sorted\_list[mid] != number:*

*if number < sorted\_list[mid]:*

*end = mid -1*

*else:*

*beg = mid + 1*

*mid = int( (beg + end)/2 )*

*if sorted\_list[mid] == number:*

*# number Found*

*return mid*

*else:*

*# number Not Found*

*return None*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

**Output:**

List = [12, 34, 35, 42, 67, 78, 99]

item=78

result = binarySearch(List, item)

if result is None:

print(item, " not found.")

else:

print(item, " found at index ", result, ", at location",(result+1))

item=92

result = binarySearch(List, item)

if result is None:

print(item, " not found.")

else:

print(item, " found at index ", result, ", at location",(result+1))

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

***Django Models***

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_from django.db import models*

*from django.db.models.fields import TimeField, BooleanField, DateField, DateTimeField, DecimalField, TextField, FloatField, IntegerField*

*import datetime*

*# from django.contrib.auth.models import User*

*class User(models.Model):*

*name = TextField(max\_length=255, null=False)*

*userid = TextField(max\_length=255, null=False)*

*dob = DateField(null=True, blank=True)*

*# Integration information*

*status = BooleanField(default=True)*

*# Time Stamp information*

*created = DateTimeField(auto\_now=True)*

*updated = DateTimeField(auto\_now\_add=True)*

*class UserAddress(models.Model):*

*user = models.ForeignKey(User, to\_field='id', related\_name='user\_addresses', on\_delete=models.CASCADE,)*

*address = TextField(max\_length=255, null=False)*

*# Time Stamp information*

*created = DateTimeField(auto\_now=True)*

*updated = DateTimeField(auto\_now\_add=True)*

*class UserRemark(models.Model):*

*# remark = models.OneToOneField(User, on\_delete=models.CASCADE)*

*user = models.OneToOneField(User, on\_delete=models.CASCADE) #, to\_field='id',related\_name='user\_remark', unique\_key = True, on\_delete = models.CASCADE)*

*# user = models.ForeignKey(User, to\_field='id', primary\_key = True, on\_delete=models.CASCADE)*

*userremark = TextField(max\_length=255, null=False)*

*# Time Stamp information*

*created = DateTimeField(auto\_now=True)*

*updated = DateTimeField(auto\_now\_add=True)*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

***Max Heap***

***SOLUTION 1***

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*import heapq*

*result = []*

*for user in User.objects.all():*

*result.append((user.id\*(-1),user.name, user.dob))*

*result*

*heapq.heapify(result)*

*result*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

***OUTPUT***

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*>>> result*

*[(-1, 'Vaidehi', datetime.date(2012, 8, 8)), (-2, 'Sarita', datetime.date(1981, 8, 17)), (-3, 'Bhushan', datetime.date(1980, 8, 22))]*

*>>> heapq.heapify(result)*

*>>> result*

*[(-3, 'Bhushan', datetime.date(1980, 8, 22)), (-2, 'Sarita', datetime.date(1981, 8, 17)), (-1, 'Vaidehi', datetime.date(2012, 8, 8))]*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

***SOLUTION 2 WITH OUTPUT***

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*>>> result1 = User.objects.all().annotate(id1=F('id')\*(-1)).values('id1')*

*>>> result1*

*<QuerySet [{'id1': -1}, {'id1': -2}, {'id1': -3}]>*

*>>> result2 = [tuple(i.values()) for i in result1]*

*>>> heapq.heapify(list(result2))*

*>>> result2*

*[(-1,), (-2,), (-3,)]*

*>>> heapq.heapify(result2)*

*>>> result2*

*[(-3,), (-2,), (-1,)]*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

***Instruction to Install and Run package***

1. **Setting up a new environment:**

Before we do anything else we'll create a new virtual environment, using venv. This will make sure our package configuration is kept nicely isolated from any other projects we're working on.

* 1. **To create a new environment:**

*python3 -m venv <env\_name>*

* 1. **Activate created environment:**

*source <env\_name>/bin/activate*

1. **Copy the package:**

Now that we're inside a virtual environment, we can copy our package.git clone

* 1. **Git Clone:**

*git clone http://<user\_name>@gitlab.cse.iitb.ac.in/saritat/sampletest.git*

* 1. **Change Directory:**

*cd sampletest/*

* 1. **Checkout Git branch**

*git checkout master*

1. **Package Settings:**
   1. **In edxapp MySQL Database:**
      1. Create Database:

create database <***db\_name>***;

* + 1. Use *'*sampletest*'* database:

*mysql -u root -p*

use <***db\_name>****;*

* + 1. Create a user with all privileges granted:

*use mysql;*

*create user '<db\_user\_name>'@'<api\_server\_ip>' IDENTIFIED BY '<db\_password>';*

*GRANT ALL PRIVILEGES ON \*.\* TO '<db\_user\_name>'@'<api\_server\_ip>' WITH GRANT OPTION ;*

*FLUSH PRIVILEGES;*

* 1. ***Change database setting of the package:***

We'll need to edit our mysql user credential to ‘DATABASES’ dictionary . Let's edit the './*sampletest*/settings.py' file:

*vi ./sampletest/settings.py*

**For example:**

*DATABASES = {*

*'default': {*

*'ENGINE': 'django.db.backends.mysql',*

*'NAME':* ***<db\_name>****,*

*'USER': '****<db\_user\_name>****',*

*'PASSWORD':'****<db\_password>****',*

*'HOST':'****<MySQL\_IP>****',*

*'PORT':'****<MySQL\_PORT>****',*

*}*

*}*

*Run django migration commands or restore the Mysql Database dump which is available in a “sampletest/db\_backup.sql” file.*

* 1. ***Package Server Setting:***

*We'll need to add our server IP to 'ALLOWED\_HOSTS' list. Let's edit the '*sampletest*/settings.py' file:*

*vi* sampletest*/settings.py*

**For example:**

ALLOWED\_HOSTS = ['127.0.0.1']

1. ***Install package requirements*:**

We can install our package requirements inside a virtual environment.

*pip install -r requirement.txt*

1. **Run server:**

Let us start up the development web server:

*python manage.py runserver <server\_IP>:<port>*