6406532306650. ✔ Quick select algorithm is an example of the divide-and-conquer approach to solving problems

6406532306651. ✖ Using Fast Select (Quick Select using MoM for pivot selection) strategy, the worst-case running time will be $O(n \log n)$.

AppDev1

Section Id :	64065348506
Section Number :	8
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	15
Number of Questions to be attempted :	15
Section Marks :	50
Display Number Panel :	Yes
Group All Questions :	No
Enable Mark as Answered Mark for Review and Clear Response :	Yes
Maximum Instruction Time :	0
Sub-Section Number :	1
Sub-Section Id :	640653100831
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Number : 113 Question Id : 640653689525 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 0

Question Label : Multiple Choice Question

THIS IS QUESTION PAPER FOR THE SUBJECT "DIPLOMA LEVEL : MODERN APPLICATION DEVELOPMENT I (COMPUTER BASED EXAM)"

ARE YOU SURE YOU HAVE TO WRITE EXAM FOR THIS SUBJECT?

CROSS CHECK YOUR HALL TICKET TO CONFIRM THE SUBJECTS TO BE WRITTEN.

(IF IT IS NOT THE CORRECT SUBJECT, PLS CHECK THE SECTION AT THE TOP FOR THE SUBJECTS REGISTERED BY YOU)

Options :

6406532306660. ✔ YES

6406532306661. ✖ NO

Sub-Section Number :	2
Sub-Section Id :	640653100832
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 114 Question Id : 640653689526 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4.5

Question Label : Multiple Choice Question

Consider the schema for the 'student' table created in SQLite database using flask-sqlalchemy.

```
CREATE TABLE "student" (
      "s_id"      INTEGER,
      "roll_number"     TEXT NOT NULL UNIQUE,
      "first_name"      TEXT NOT NULL,
      "last_name" TEXT NOT NULL,
      PRIMARY KEY("s_id" AUTOINCREMENT)
);
```

What will be the output of the flask_sqlalchemy command given below?

```
>>> s1 = Student(roll_number = "M01", first_name = "John", last_name = "Doe")
>>> db.session.add(s1)
>>> s2 = Student(roll_number = "M02", first_name = "John", last_name = "Luther")
>>> db.session.add(s2)
>>> s3 = Student(roll_number = "M03", first_name = "Harry", last_name = "Doe")
>>> db.session.add(s3)
>>> db.session.commit()
>>> user1= Student.query.filter_by(first_name="John").first()
>>> user1.first_name= "Harry"
>>> user1.last_name= "Luther"
>>> db.session.commit()
>>> s1 = Student.query.all()
>>> for student in s1:
...    print(student.first_name)
...    print(student.last_name)
...
```

Options :

6406532306662. ✖

```
Harry
Luther
Harry
Luther
Harry
Doe
```

6406532306663. ✔

```
Harry
Luther
John
Luther
Harry
Doe
```

6406532306664.

```
Harry
Luther
Harry
Doe
Harry
Doe
```

```
Harry
Doe
John
Luther
Harry
Doe
```

6406532306665.

Question Number : 115 Question Id : 640653689543 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4.5

Question Label : Multiple Choice Question

Consider the following flask application.

```python
from flask import Flask, abort
app = Flask(__name__)

users = ["admin", "user_1", "user_2", "user_3"]

@app.route('/home/<string:username>', methods = ['POST'])
def home(username):
    if username in users:
        return f"<h1>Hello {username}, Welcome!</h1>"
    else:
        abort(404)

@app.errorhandler(404)
def user_error_1(error):
    return "<h1>The user you are looking for is invalid.</h1>"

@app.errorhandler(405)
def user_error_2(error):
    return "<h1>Please check the web request.</h1>"

app.run()
```

If the application is running locally on `http://127.0.0.1:5000`. What will be rendered on the browser for url `http://127.0.0.1:5000/home/user_4` ?

Options :

6406532306722. ✖ **Hello user_4, Welcome!**

6406532306723. ✖ **The user you are looking for is invalid.**

6406532306724. ✖ **Not Found**

6406532306725. ✔ **Please check the web request**

Sub-Section Number :	3
Sub-Section Id :	640653100833
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 116 Question Id : 640653689527 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3 Max. Selectable Options : 0

Question Label : Multiple Select Question

Consider the following flask app. Given that `test_request_context()` allows text to be printed on the terminal, which of the following statements is/are correct?

```
from flask import Flask, url_for
app = Flask(__name__)

@app.route('/home')
def index():
    return 'Mad-I Student Data'

@app.route('/student/<student_name>/<int:student_id>')
def profile(student_name,student_id):
    return f'Student Name:{student_name},Student ID:{student_id}'

with app.test_request_context():
    #== print statement ==#
```

Options :

6406532306666. ✖ If `#== print statement ==#` is replaced by;
`print(url_for('index'))`,
the output on the terminal will be;
`Mad-I Student Data`

6406532306667. ✖ If `#== print statement ==#` is replaced by;
`print(url_for('profile', student_name='John', student_id= 101))`,
the output on the terminal will be;
`/student?student_name=John&student_id=101`

6406532306668. ✔ If `#== print statement ==#` is replaced by;
`print(url_for('index'))`,
the output on the terminal will be;
`/home`

6406532306669.

If `#== print statement ==#` is replaced by;
`print(url_for('profile',student_name='John',student_id=101))`,
the output on the terminal will be;
✔ `/student/John/101`

Question Number : 117 Question Id : 640653689529 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3 Max. Selectable Options : 0

Question Label : Multiple Select Question

Consider the following flask application.

app.py

```
from flask import Flask, abort

app = Flask(__name__)

@app.route('/<uname>')
def index(uname):
      if uname[0].isdigit():
            abort(400,"Bad Request")
      return '<h1>Good Username</h1>'

app.run(debug=True)
```

Which of the following statements is/are true if the application is running locally on http://127.0.0.1:5000 ?

Options :

6406532306674. ✖ For URL `http://127.0.0.1:5000/MAD1?uname=102` the browser will render Bad Request.

6406532306675. ✔ For URL `http://127.0.0.1:5000/MAD1?uname=102` the browser will render **Good Username**.

6406532306676. ✔ For URL `http://127.0.0.1:5000/102?uname=MAD1` the browser will render Bad Request.

For URL http://127.0.0.1:5000/102?uname=MAD1 the browser will render

6406532306677. ✖ **Good Username.**

Sub-Section Number : 4

Sub-Section Id : 640653100834

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 118 Question Id : 640653689528 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

In the context of in-memory databases, what is the primary function of an index?

Options :

6406532306670. ✖ To encrypt sensitive data

6406532306671. ✔ To optimize query performance by speeding up data retrieval

6406532306672. ✖ To manage data replication

6406532306673. ✖ To store metadata about the database structure

Question Number : 119 Question Id : 640653689536 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

Consider the following python code snippet.

```
from string import Template as T1
from jinja2 import Template as T2
temp = "A student needs to complete $c1 level to get to the {{c2}}
level."

temp = T1(temp)
out = temp.substitute({'c1':'foundation', 'c2':'diploma'})
print(out)
out= T2(out)
print(out.render({'c2':'diploma'}))
```

What is the generated output on python console?

Options :

6406532306694. ✖ A student needs to complete foundation level to get to the diploma level.

A student needs to complete foundation level to get to the diploma level.

6406532306695. ✖ A student needs to complete $c1 level to get to the {{c2}} level.

A student needs to complete foundation level to get to the diploma level.

6406532306696. ✔ A student needs to complete foundation level to get to the {{c2}} level.

A student needs to complete foundation level to get to the diploma level.

6406532306697. ✖ A student needs to complete foundation level to get to the {{diploma}} level.

A student needs to complete foundation level to get to the diploma level.

Question Number : 120 Question Id : 640653689538 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

In the context of API documentation, what is typically provided to describe the available endpoints, request parameters and response formats?

Options :

6406532306702. ✔ API documentation or API reference

6406532306703. ✖ API key

6406532306704. ✖ Software source code

6406532306705. ✖ OAuth tokens

Sub-Section Number : 5

Sub-Section Id : 640653100835

Question Shuffling Allowed : No

Is Section Default? : null

Question Id : 640653689530 Question Type : COMPREHENSION Sub Question Shuffling Allowed : No Group Comprehension Questions : No Question Pattern Type : NonMatrix Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Question Numbers : (121 to 122)

Question Label : Comprehension

Consider the models "Channel" and "Video" used to create the tables "channel" and "video" in SQLite database using flask-sqlalchemy and answer the given subquestions.

```
class Channel(db.Model):
    id = db.Column(db.Integer, primary_key = True)
    name = db.Column(db.String(50), unique = True, nullable = False)
    videos = db.relationship("Video", backref = "playlist")

class Video(db.Model):
    id = db.Column(db.Integer, primary_key = True)
    name = db.Column(db.String(50), unique = True, nullable = False)
    channel = db.Column(db.Integer, db.ForeignKey("channel.id"))
```

Sub questions

Question Number : 121 Question Id : 640653689531 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

What relationship do the tables "channel" and "video" share.

Options :

6406532306678. ✖ One-to-one

6406532306679. ✔ One-to-many

6406532306680. ✖ Many-to-one

6406532306681. ✖ Many-to-many

Question Number : 122 Question Id : 640653689532 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3 Max. Selectable Options : 0

Question Label : Multiple Select Question

If "ch1" is a channel object created in "channel" table with primary key 1, then the correct way(s) of creating a video object "vid1" in the video table that belongs to "ch1" object is/are _____. (let "name" of the video be "Intro to Python")

Options :

6406532306682. ✖
```
>>> vid1 = Video(name = "Intro to Python", channel = ch1)
>>> db.session.add(vid1)
>>> db.session.commit()
```

6406532306683. ✔
```
>>> vid1 = Video(name = "Intro to Python", channel = 1)
>>> db.session.add(vid1)
>>> db.session.commit()
```

6406532306684. ✔
```
>>> vid1 = Video(name = "Intro to Python", playlist = ch1)
>>> db.session.add(vid1)
>>> db.session.commit()
```

6406532306685. ✖
```
>>> vid1 = Video(name = "Intro to Python", playlist = 1)
>>> db.session.add(vid1)
>>> db.session.commit()
```

Sub-Section Number :	6
Sub-Section Id :	640653100836
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Id : 640653689533 Question Type : COMPREHENSION Sub Question Shuffling Allowed : No Group Comprehension Questions : No Question Pattern Type : NonMatrix Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Question Numbers : (123 to 124)

Question Label : Comprehension

Consider the following flask application running locally on `http://127.0.0.1:5000` and the table `"student"` in the database and answer the given subquestions.

id	name	roll	course
Filter	Filter	Filter	Filter
1	Sandeep	STU101	History
2	Ramesh	STU102	Physics
3	Aarav	STU103	Chemistry
4	Meera	STU104	Biotechnology
5	Ramesh	STU105	Mechanics
6	Rishav	STU106	Mathematics

Filename: app.py

```
parser = reqparse.RequestParser()
parser.add_argument('roll')
parser.add_argument('name')
parser.add_argument('course')
class TestApi(Resource):
    def get(self, name):
        stud = Student.query.filter_by(name = name).first()
        return {
            "roll_no": stud.roll
        }

    def post(self):
        stud = parser.parse_args()
        newStud = Student(roll = stud['roll'], name = stud['name'],
                          course = stud['course'])
        db.session.add(newStud)
        db.session.commit()
        return '', 201

    def put(self, roll, course):
        stud = Student.query.filter_by(roll = roll).first()
        stud.course = course
        db.session.commit()
        return ''
  api.add_resource(TestApi,
                   "/api/<name>",
                   "/student/",
                   "/app/<roll>/<course>")
```

Sub questions

Question Number : 123 Question Id : 640653689534 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Map the api resources in column A with the correct URLs given in column B.

Column A	Column B
1. GET	a. http://127.0.0.1:5000/student
2. PUT	b. http://127.0.0.1:5000/api/Meera
3. POST	c. http://127.0.0.1:5000/app/STU103/Sociology

Options :

6406532306686. ✖ 1 → c, 2 → a, 3 → b

6406532306687. ✖ 1 → a, 2 → b, 3 → c

6406532306688. ✖ 1 → b, 2 → a, 3 → c

6406532306689. ✔ 1 → b, 2 → c, 3 → a

Question Number : 124 Question Id : 640653689535 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

If the app first receives the request,

```
curl http://127.0.0.1:5000/app/STU105/Robotics -X PUT
```

what will be the final output on the shell after writing the following lines of code one below the other in the Python shell?

```
>>> from app import *
>>> students = Student.query.filter_by(name = "Ramesh").all()
>>> for student in students:
...     print(student.course)
```

Options :

6406532306690. ✖ Physics
Physics

6406532306691. ✖ Robotics
Robotics

6406532306692. ✔ Physics
Robotics

6406532306693. ✖ Robotics
Physics

Sub-Section Number : 7

Sub-Section Id : 640653100837

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 125 Question Id : 640653689537 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider a client which is located 3000 km from the server makes a request through the cable. Suddenly after the request reaches the server, the cable breaks and the response is now to be sent to the client via air. This change of medium caused an additional delay of 30 ms at the server end. How long will the client have to wait for receiving the response? (speed on cable = 2e8 m/s and in air 3e8 m/s)

Options :

6406532306698. ✔ 55 milliseconds

6406532306699. ✖ 50 milliseconds

6406532306700. ✖ 80 milliseconds

6406532306701. ✖ 60 milliseconds

Question Number : 126 Question Id : 640653689539 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

The throughput of a Solid State Device (SSD) is 180 MB/sec. If it is to be replaced by a standard HDD whose lens can read data with the rate of 4.8 Megabits/revolution. What should be the speed of rotation (in RPM) of HDD to get a throughput equal to that of the SDD?

Options :

6406532306706. ✖ 300

6406532306707. ✖ 900

6406532306708. ✖ 12000

6406532306709. ✔ 18000

Question Number : 127 Question Id : 640653689542 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

What will be the output of the following Python code snippet?

```
def modify(func):
    def wrapper(n):
        mylist = []
        for i in range(1, n + 1):
            if n%i == 0:
                mylist.append(i)
        return mylist
    return wrapper

@modify
def myFunc(n):
    mylist = []
    for i in range(1, n + 1):
        if i%2 == 0:
            mylist.append(i)
    return mylist

print(myFunc(14))
```

Options :

6406532306718. ✖ `[1, 3, 5, 7, 9, 11, 13]`

6406532306719. ✖ `[2, 4, 6, 8, 10, 12, 14]`

6406532306720. ✔ `[1, 2, 7, 14]`

6406532306721. ✖ `[1, 3, 5, 7, 11, 13]`

Sub-Section Number :	8
Sub-Section Id :	640653100838
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 128 Question Id : 640653689540 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4.5 Max. Selectable Options : 0

Question Label : Multiple Select Question

Consider the following flask view function defined to add a product and select correct statement(s).

```
@app.route("/add", methods = ['GET', 'POST'])
def add_product():
    if request.method == 'POST':
        name = request.form.get('name')
        category = request.form.get('category')
        if name == "":
            return redirect('/add')
        if category == "":
            return redirect('/add')
        return f"Product {name} is added in {category} category."

    return """
        <form action="/add" method="post">
         Product Name: <input type="text" name="name"
         minlength="2"><br>
         Product Category: <input type="text" name="category"
         required><br>
         <input type="submit" value="Add Product">
        </form>
        """
```

Options :

6406532306710. ✖ The form to add the product is statically generated.

6406532306711. ✔ The form to add the product is dynamically generated.

6406532306712. ✔ The form rendered for `/add` endpoint has frontend validation.

6406532306713. ✔ The form rendered for `/add` endpoint has backend validation.

Question Number : 129 Question Id : 640653689541 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction

Time : 0

Correct Marks : 4.5 Max. Selectable Options : 0

Question Label : Multiple Select Question

Consider the following Python code snippet.

Filename: log.py

```
import logging
import sys

logging.basicConfig(level=logging.WARNING,
                    format='%(levelname)s - %(message)s')

num1, num2 = int(sys.argv[1]), int(sys.argv[2])

if num2 > 0:
    logging.debug(f"Division {num1}/{num2} is possible")
else:
    logging.critical("FATAL - Division by zero is not possible")
```

Which of the following statements is/are correct?

Options :

6406532306714. ✖

For the command: `python log.py 12 24`
The output on the terminal will be:

```
DEBUG - Division 12/24 is possible
```

6406532306715. ✖

For the command: `python log.py 0 24`
The output on the terminal will be:

```
DEBUG - Division 0/24 is possible
```

6406532306716. ✔

For the command: `python log.py 15 0`
The output on the terminal will be:

```
CRITICAL - FATAL - Division by zero is not possible
```

6406532306717. ✔

For the command: `python log.py 0 0`
The output on the terminal will be:

```
CRITICAL - FATAL - Division by zero is not possible
```

MLF

Section Id :	64065348507
Section Number :	9
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	11
Number of Questions to be attempted :	11
Section Marks :	40
Display Number Panel :	Yes
Group All Questions :	No
Enable Mark as Answered Mark for Review and Clear Response :	Yes
Maximum Instruction Time :	0
Sub-Section Number :	1
Sub-Section Id :	640653100839
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Number : 130 Question Id : 640653689544 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 0

Question Label : Multiple Choice Question

THIS IS QUESTION PAPER FOR THE SUBJECT "DIPLOMA LEVEL : MACHINE LEARNING FOUNDATIONS (COMPUTER BASED EXAM)"

ARE YOU SURE YOU HAVE TO WRITE EXAM FOR THIS SUBJECT?