PURE

Progressive
Universal
Respectful
Eccentric

2023 Apache AGE Internship Program (Egypt)

July 2023

Apache AGE (Agens Graph Extension)

The 1st Apache Open-Source Top-Level Project (TLP) integrating Relational Database and Graph technologies

https://age.apache.org/

https://github.com/apache/age (Graph Database)

https://github.com/apache/age-viewer (Graph Data Viewer)

- Apache AGE is a PostgreSQL extension that adds graph query functionality to PostgreSQL.

- Through using Graph technologies such as the openCypher query specification and Graph-based data analysis algorithms, users can access, store, query and analyze graph data stored in relational database such as PostgreSQL.

Since 2020, Bitnine Global has been the major contributor of the Apache AGE project since Bitnine donated AGE to the Apache Software Foundation, the world's largest open source foundation

Established in 1999, The Apache Software Foundation is the world's largest open source foundation, stewarding 227M+ lines of code and providing more than $22B+ worth of software to the public.

Together, Let's Innovate How Data is Stored, Viewed and Analyzed
By Developing Graph Technologies for all Relational Databases

For 6 months, the interns will be given a chance to gain a deeper understanding on **how database system works, Graph data, queries and algorithms, and practical experience with developing graph database management system software and/or its viewer.**

- Apache AGE (PostgreSQL extension – C programming, PostgreSQL core, openCypher), https://github.com/apache/age
- Apache AGE Viewer (Graph data viewer – Node.js, React, Go, Graph visualization), https://github.com/apache/age-viewer
- Graph-oriented data analysis algorithms – Fortran, Python, PL/Python, PL/SQL, …

and also a job offer and/or official Apache Committer status based on contributions/performance.

The 2022-23 Apache AGE Internship Program (Egypt)
- Starts once the hiring is done
- Pay is **(4)** USD an hour
- Mostly remote with occasional visits to meetups
- 20 hours a week, could vary depending on where you are, your current status and the tasks allocated
- Technical Mentors : Bitnine's Apache AGE Project Members, PostgreSQL core contributors and Graph data scientists

Those who wish to continue with the application for the (paid) internship program

Please send us your latest CV or resume.

Please participate in the coding test as described in the later slides.

All emails and inquiries should be sent to global.hr@bitnine.net

Coding Test Questions

Question 1 and 2 (Backend)
- Developing Apache AGE requires a lot of logical thinking that backend development often requires.

Question 3 (Database Driver / Interface)
- Database driver is a piece of software that allows applications to connect to and interact with the database system.

Question 4 (Frontend)
- Graph visualization is an integral part of any Graph database system software offering.

Question 5 (Frontend/Backend/Database)
- Developing Apache AGE requires a lot of logical thinking that Frontend/backend development often requires.

Question 6 (Basic Algos)
- Apache AGE requires a lot of logical thinking that backend development often requires.

Please be advised that you can attempt this coding test as follows:
1) Both Question 1 and Question 2
   OR
2) Both Question 3 and Question 4
   OR
3) Both Question 5(5-1, 5-2) and Question 6(6-1, 6-2, 6-3)

[Question No. 1]

Node is defined as follows :

typedef struct Node
{
    TypeTag type;
} Node;
typedef enum TypeTag {
...
}

Using this structure, please write a function that returns fibonacci sequence based on the following arithmetic operations (+, -, *, /) and conditions.
The fibonacci function should be implemented using Dynamic Programming.

```
main()
{
   Node *add = (*makeFunc(ADD))(10, 6);
   Node *mul = (*makeFunc(MUL))(5, 4);
   Node *sub = (*makeFunc(SUB))(mul, add);
   Node *fibo = (*makeFunc(SUB))(sub, NULL);

   calc(add);
   calc(mul);
   calc(sub);
   calc(fibo)
}

Output
add : 16
mul : 20
sub : -4
fibo : 2
```

Please submit

- Source code (**Please write this in C**)

- README text file explaining development

  environment and how to compile and run the

  source code.

[Question No. 2]

- Implement the following piecewise recurrence relation in the 3 different ways.

- Explain the differences (advantages, disadvantages) between the 3.

- Must be written in C

The problem

F(n) = F(n-3) + F(n -2) where F(0) = 0, F(1) = 1, and F(2) = 2.
Assume that n will be less than or equal to the maximum integer value and non-negative. You only need to write the function(s).

Please submit

- Source code (**Please write this in C**)

- README text file explaining development environment and how to compile and run the source code.

[Question No. 3]

The open-source PostgreSQL drivers are available for various programming languages, including C++, Java and Node.js.
Using the language that you are most confident in, please modify the PostgreSQL driver source to return outputs in JSON string format (like the following).
Please submit the source code, driver file and an instruction for using the driver.

```
{
    "status_code" : 200
    ,"data" : [
        {"user_id":1,"name":"John","age":28}
        ,{"user_id":2,"name":"Tom", "age":29," phone":"1-800-123-1234"}
        ,{"user_id":3,"name":"Jenny", "age":34}
}
```

```
# Postgresql table
CREATE TABLE IF NOT EXISTS public.user_table
(
user_id numeric(10,0) NOT NULL,
name character varying(50) COLLATE pg_catalog."default" NOT NULL,  age numeric(3,0) NOT NULL,
phone character varying(20) COLLATE pg_catalog."default",
CONSTRAINT user_table_pkey PRIMARY KEY (user_id)
);
INSERT INTO public.user_table (user_id, name, age, phone) VALUES (3, 'Jenny', 34, NULL);
INSERT INTO public.user_table (user_id, name, age, phone) VALUES (2, 'Tom', 29, '1-800-123-1234');
INSERT INTO public.user_table (user_id, name, age, phone) VALUES (1, 'John', 28, NULL);
```

Please submit

- Source code

- Driver file

- README text file explaining development

  environment and how to use the driver

- **Please attempt this using both Go, Python**

[Question No. 4]

- Fork the Apache AGE Viewer project (to your own GitHub repo) and further develop Apache AGE Viewer (using React and other front-end frameworks) to make it more user friendly

  https://github.com/apache/age-viewer

Please submit

- A link to your GitHub repo where your work on Apache AGE Viewer exists
- README explaining your version of Apache AGE Viewer

[Question No. 5]

Question 5 - 1:

- Need to implement a login/signup page that will allow users to view some content after signing up and then login.

Question 5 - 2:

- Design a webpage similar to the Bitnine's official website https://bitnine.net/

(You can also implement this in first question code)

[Question No. 6]

Question 6-1:

- Write a function that will take input from the user and check if it is palindrome, if the word is not palindrome then return it in the reverse order. For example "hello" is not a palindrome, it should return "olleh"

Question 6-2:

- Let's say we have an array of first 45 numbers with one missing number. Need to find a missing integer from the list (JavaScript)

[Question No. 6]

Question 6-3:

Complete the following code to make a <u>Javascript</u> based digital clock.

```
function myDigitalClock(){
    var d1 = new Date();
    var hours = date.getHours(); // 0 - 23
    var minutes = date.getMinutes(); // 0 - 59
    var seconds = date.getSeconds(); // 0 - 59
    var zone = "AM";

//Write the code here
//
//

}
myDigitalClock();
```