



De Thi Cuoi Ky LTM Đề 2 - 0000

Lập trình mạng (Trường Đại học Bách Khoa - Đại học Đà Nẵng)



Scan to open on Studocu

ĐỀ THI VÀ BÀI LÀM - Đề 2

Tên học phần: Lập Trình Mạng

Mã học phần: Hình thức thi: *Tự luận có giám sát*

Thời gian làm bài: 60 phút (*không kể thời gian phát đề và nộp bài*)

Được sử dụng tài liệu khi làm bài. Không chia sẻ bài cho nhau, nếu phát hiện sẽ chia đều số điểm.

Họ tên:...Le Thanh Viet.....**Lớp:**...21TCLC_DT4.....**MSSV:**.....
102210287.....

Sinh viên làm bài trực tiếp trên tệp này, lưu tệp với định dạng MSSV_HọTên.pdf và nộp bài thông qua MSTeam:

Câu 1 (3 điểm): Hãy viết chương trình theo giao thức TCP với các chức năng sau:

a) Client:

- a. Kết nối tới server
- b. Tạo 1 số ngẫu nhiên n trong đoạn từ 2 đến 10^9
- c. Lặp đi lặp lại các bước sau:
 - i. Gửi cho server 1 chuỗi thể hiện số nguyên n đã tạo.
 - ii. Nhận kết quả trả về từ server là 1 chuỗi thể hiện số nguyên m . Gọi t là số lượng số nguyên số cùng nhau với n trong đoạn [1, n]. Kiểm tra xem tất cả các số từ $m^1 \% n$ đến $m^t \% n$ có số nào trùng nhau hay không. Nếu có thì in ra "**số m gửi về không hợp lệ**" quay lại bước **i**. Ngược lại thì in "**Số m gửi về hợp lệ**", gửi cho server số "1" và nhảy tới bước **d**.

Ví dụ: $n = 14$, thì ta có $t = 6$ vì chỉ có 1,3, 5, 9, 11, 13 là nguyên tố cùng nhau với 14. Nếu m trả về là 2 thì $2^1 \% 14 == 2^4 \% 14$. Cần in ra "**số 2 gửi về không hợp lệ**". Nhưng nếu m trả về là 3 thì $3^1 \% 14 != 3^2 \% 14 != 3^3 \% 14 != 3^4 \% 14 != 3^5 \% 14 != 3^6 \% 14$, vậy cần in ra "**Số 3 gửi về hợp lệ**". Gửi cho server số "1".

d. Đóng kết nối

b) Server:

- a. Lắng nghe và chấp nhận kết nối từ các client. Ứng với mỗi kết nối tạo 1 luồng xử lý sau:

i. Lặp đi lặp lại các bước sau **không quá 10 lần**:

Bước 1. Nhận 1 chuỗi thể hiện số nguyên n từ client gửi tới.

Bước 2. Nếu $n == 1$ thì nhảy tới **ii**

Bước 3. Hãy tìm hoặc đoán số nguyên m và gửi cho client số đó.

Bước 4. Quay về bước 1.

ii. Đóng kết nối.

Yêu cầu, mỗi lần client gửi 1 chuỗi cho server, server phải phản hồi client trong vòng không quá 1 giây. Nếu vòng lặp server chạy 10 lần vẫn không tìm hoặc đoán được số nguyên m thì không được điểm tối đa.

**# Trả lời:
Dán code server vào bên dưới**

```
public class Server {
    private static final int SERVER_PORT = 12345;
    private static final int MAX_ATTEMPTS = 10;

    public static void main(String[] args) {
        ExecutorService executorService = Executors.newCachedThreadPool();

        try (ServerSocket serverSocket = new ServerSocket(SERVER_PORT)) {
            while (true) {
                Socket clientSocket = serverSocket.accept();
                executorService.execute(new ClientHandler(clientSocket));
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    private static class ClientHandler implements Runnable {
        private final Socket clientSocket;

        public ClientHandler(Socket clientSocket) {
            this.clientSocket = clientSocket;
        }

        @Override
        public void run() {
            try (PrintWriter out = new PrintWriter(clientSocket.getOutputStream(),
true);
                 BufferedReader in = new BufferedReader(new
InputStreamReader(clientSocket.getInputStream()))) {

                Random random = new Random();
                int attempts = 0;

                while (attempts < MAX_ATTEMPTS) {
                    String input = in.readLine();
                    int n = Integer.parseInt(input);

                    if (n == 1) {
                        continue;
                    }

                    int m = 2;
                    out.println(m);

                    attempts++;
                }

                clientSocket.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

Dán code client vào bên dưới

```
public class Client {
    private static final String SERVER_ADDRESS = "localhost";
    private static final int SERVER_PORT = 12345;
```

```

public static void main(String[] args) {
    try (Socket socket = new Socket(SERVER_ADDRESS, SERVER_PORT);
        PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
        BufferedReader in = new BufferedReader(new
InputStreamReader(socket.getInputStream()))) {

        Random random = new Random();
        int n = 14;

        while (true) {
            out.println(n);
            String response = in.readLine();
            int m = Integer.parseInt(response);

            int t = 0;
            for (int i = 1; i <= n; i++) {
                if (gcd(i, n) == 1) {
                    t++;
                }
            }

            boolean isValid = true;
            for (int i = (m * m) % n; i <= (m + 10) % n; i++) {
                if (i % n == m) {
                    isValid = false;
                    break;
                }
            }

            if (isValid) {
                System.out.println("Sô " + m + " gửi về hợp lệ");
                out.println("1");
                break;
            } else {
                System.out.println("Sô " + m + " gửi về không hợp lệ");
            }
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}

private static int gcd(int a, int b) {
    if (b == 0) {
        return a;
    }
    return gcd(b, a % b);
}

```

Dán các kết quả thực thi vào bên dưới

```
/Users/thanhviet/Library/Java/JavaVirtualMachines/openjdk-21.0.1/Contents/Home/bin/  
Số 3 gửi về hợp lệ
```

```
Process finished with exit code 0
```

Câu 2 (2 điểm): Hãy cho biết nếu chuyển chương trình ở câu 1 qua giao thức UDP thì các công việc cần làm gồm những gì? (Không cần code, chỉ cần mô tả)

Trả lời:

-Trong giao thức TCP, chúng ta sử dụng luồng đọc (BufferedReader) và ghi (PrintWriter) để truyền và nhận dữ liệu qua kết nối đã thiết lập. Trong khi đó, trong giao thức UDP, chúng ta sử dụng gói dữ liệu (DatagramPacket) và socket (DatagramSocket) để gửi và nhận dữ liệu

Tạo DatagramSocket để nhận và gửi dữ liệu

Tạo DatagramPacket có chứa mảng byte để đựng dữ liệu

Sử dụng DatagramPacket và DatagramSocket.send để gửi dữ liệu đi

Câu 3 (5 điểm): Trong phần bài tập JSP/Servlet đã nộp

Đề bài tập: Có 1 tính toán lớn (chạy ngầm, ví dụ như xử lý dữ liệu lớn, xử lý video, convert pdf qua doc, xây dựng mô hình học máy, kiểm tra đạo code, đạo văn tự động,...). Khi client gửi thông tin cần thực hiện xử lý, server sẽ đẩy thông tin đó vào 1 hằng đợi để thực hiện. Client sẽ xem kết quả xử lý thông qua account của bản thân.

Hãy trả lời các câu hỏi sau:

a) Cho biết tên của các thành viên trong nhóm, kể cả bản thân

Trả lời: viết câu trả lời vào bên dưới

Lê Thanh Việt
Hoàng Chiến Thắng

b) Mô tả chức năng chính mà bản thân đã đóng góp vào chương trình.

Trả lời: viết câu trả lời vào bên dưới

đăng nhập, mã hoá, upload file, convert file pdf, xây dựng database, validate dữ liệu,
xây dựng giao diện

c) Hãy mô tả cách thức để có thể xử lý một request với thời gian lớn mà không bị “request time out”. Hãy trích tối đa 10 dòng code trong bài tập đã làm thể hiện điều đó! (có thể thêm comment nếu cần, nhưng không được trích quá 10 dòng code)

Trả lời: viết câu trả lời vào bên dưới

```
if (requestQueue.size() > 0) {  
    // Bắt đầu một luồng mới cho yêu cầu tiếp theo  
    Thread t = new Thread(new UploadFileBO(firstRequest, firstUser));  
    t.start();  
}
```

Đà Nẵng, ngày 12 tháng 12 năm 2023