

CECS 327  
Assignment 9  
Total: 30 Points

---

General Instruction

- Submit your work in the Dropbox folder via BeachBoard (Not email or in class).
- 

1. (5 points) Fill out the following `google sheet`.

- **Due by the Lab session (4/16 for S3, 4/17 for S5)**
- Click here to visit the `google sheet`
- Find your CSULB ID.
- Note your Pastry ID.
- A Pastry ID (node ID) is a quaternary number, i.e., base-4 numeral system.
- Write down your AWS IPv4 address into the `google sheet`.

2. (25 points) Implement a Pastry routing server in Java on your AWS server.

- Read the section 10.5.1 carefully.
- Use the Pastry IDs and IP addresses of the `google sheet`.
- Build your own **leaf set** table as shown in Table 1.

Table 1: Pastry leaf set ( $l = 2$ ) for Pastry ID: 1230. Each cell contains ID:IP.

SMALLER		LARGER	
1220:x.x.x.x	1223:x.x.x.x	1232:x.x.x.x	1300:x.x.x.x

- Build your own **routing** table as shown in Table 2.

Table 2: Pastry routing table for Pastry ID: 1230. Each cell contains ID:IP.

0-132:x.x.x.x	<b>1-230:y.y.y.y</b>	2-012:x.x.x.x	3-320:x.x.x.x
10-31:x.x.x.x	11-23:x.x.x.x	<b>12-30:y.y.y.y</b>	13-10:x.x.x.x
120-x:NULL	121-1:x.x.x.x	122-0:x.x.x.x	<b>123-0:y.y.y.y</b>
<b>1230:y.y.y.y</b>	1231:NULL	1232:x.x.x.x	1233:NULL

1. y.y.y.y is the IP address of Pastry ID: 1230.
  2. For the first, second, and third row, choose a node randomly. For instance, the **prefix 11**, select one of nodes whose ID has the **common prefix 11**. It can be one of 1100, 1101, 1102, ..., and 1133.
- Open inbound UDP port 32710 on the AWS server console.

- Refer Figure 4.3 and Figure 4.4 of the text book and the assignment 5.
- The program specification.
  1. It should use **Map** data structures to store Table 1 and Table 2.
  2. It should use **UDP datagram** (port 32710) **NOT** **TCP stream**.
  3. It should **reply** the (Node ID):(IP address) of a **request** Pastry ID.
  4. It should be able to handle request messages with white spaces. For instance, '1230 ', ' 123 ', ' 12', ... are all valid ones.
  5. It should replay 'INVALID REQUEST' when it receives request messages other than quaternary number whose length is less than or equal to 4.
  6. It should replay 'NULL' when it can not find the specified node.
  7. Request - Reply examples

```
1230 - 1230:y.y.y.y
123  - 1230:y.y.y.y
12   - 1230:y.y.y.y
1    - 1230:y.y.y.y
120  - NULL
1231 - NULL
1300 - 1300:x.x.x.x
122  - 1220:x.x.x.x
13   - 1310:x.x.x.x
2    - 2012:x.x.x.x
0123 - 0132:x.x.x.x
1030 - 1031:x.x.x.x
1123 - 1123:x.x.x.x
1210 - 1211:x.x.x.x
1211 - 1211:x.x.x.x
1212 - 1211:x.x.x.x
1213 - 1211:x.x.x.x
```
- Test the server program by implementing a simple UDP client on your local machine.
- Submit the source codes of the server and client.
- Leave your **Pastry ID** and **AWS IP** address on the comment section.
- Make sure that the server program is running on your **AWS** server after submitting your work by using the command `nohup java SERVER_PROGRAM &`.