

CSC 344 Computer Networks, Fall 2018

Programming Project

(Due: Nov. 16th(Friday) 11:59pm)

<Submission Guideline>

This project requires an individual work. Do NOT share your code with other people and I will NOT accept late submission. You have to submit **your Source Code** to Blackboard Programming Project Section before due.

Note that this project has **10%** of the total final grade. Even though your program is not working completely, it is possible to get partial credits depending on the level of the submitted work.

You may use any language (ex. C, C++, Java, Python, etc.). When you submit your work to Blackboard, include **Readme.txt** which describes how to implement your code and check the result by the code. TA will follow your **Readme file** when he grades your project. If your code is not working properly in TA's computer but your code is working well at your computer, you need to show the DEMO to TA. This description may have some typos. If you find it, let me know.

<Project Description and Requirements>

Basically, an objective of this project is to implement **CHORD** which learned in class. I recommend you study CHORD lecture note part before you start the project.

But, some additional requirements should be considered by the below description.

For your information, I define and assume some notations which we use in this project.

B: Circular ID-space with B-bit → 2^B Node ID spaces are possible.

(where, $5 \leq B \leq 10$)

N: The number of nodes in Chord.

(where, $5 \leq N \leq 15$)

<Project Scenario>

You have to follow the below scenario for your project.

Note that B, N values should be entered by the user in the result screen.

(1) From the information B, N from the user, **N nodes** are created.

Each **unique Node ID** is decided randomly from 2^B Node ID spaces.

(For example, if $B = 7$, one possible node ID is N20 (because of $20 \leq 2^7 (=128)$)

(2) Then, create a **finger table** per node.

(3) One Key ID is also decided randomly.

(For example, one possible Key ID is K70 (because of $70 \leq 2^7 (=128)$)

Also, one node is chosen from the created N nodes, which is referred as **N_{start}**

(Assume that we simply select the Node with the **SECOND smallest Node ID** from N nodes)

(4) Now, **DISPLAY** both the selected Key ID and **N_{start}** in the result screen.

Now, **N_{start}** wants to find the selected Key ID in Chord system.

(5) In the result screen, show the **Visiting Order** of Nodes to find the Key ID (ex. from **N_{start}** to the final node.)

(6) Also, in the result screen, show **the finger table of EACH visiting node** including **N_{start}**