COMP108 Data Structures and Algorithms
Lab Exercises (Week 8)
Due: 23 April 2021, 5:00pm

Information

- Submission: Submit the file COMP108W08.java to SAM https://sam.csc.liv.ac.uk/COMP/CW_Submissions.pl?qryAssignment=COMP108-18
- Submission of lab/tutorial exercises contributes to 10% of the overall module mark. Submission is marked on a pass/fail basis - you will get full marks for submitting a *reasonable attempt*.
- Late submission is **NOT** possible. Individual feedback will not be given, but solutions will be posted promptly after the deadline has passed.
- These exercises aim to give you practices on the materials taught during lectures and provide guidance towards assignments.
- Relevant lectures: **Lectures 11 & 12 and Lecture 8 (Video 2)**
- You can refer to the guidance on how to use the web-based IDE https://ide.cs50.io/.

1. **Programming — Preparation**

   (a) Download three java files "COMP108W08App.java", "COMP108W08.java" and "Node.java" from Canvas via the link "Labs & Tutorials" → "Week 8".

   (b) Compile the programs by typing first **javac COMP108W08.java** and then **javac COMP108W08App.java**. There should be two files created: COMP108W08.class and COMP108W08App.class.

   (c) Run the program by typing **java COMP108W08App**.

   (d) **Every time you have edited COMP108W08.java, you have to (i) recompile by javac COMP108W08.java and then (ii) run by java COMP108W08App.**

2. **The Scenario we work with this week is the same as that in Week 04 except that the movie database is now stored in a linked list instead of an array.**

   (a) **IMPORTANT:** In this lab, we are considering the following scenario. We have a database of movies (stored as linked list of integer ID) and some requests from customers who want to watch movies (stored as array of integer ID). We want to answer two questions:

      i. Which requested movies do not exist in the database? Report this in the same order of the requests.

      ii. For each movie in the database, how many times it has been requested by the customers? Report this in the same order of the database.

   We will need to write nested loop to answer these questions.

   (b) You are going to work on COMP108W08.java to complete two methods: `notExists()` for Task 1 and `count()` for Task 2.

3. **Programming with Nested Loops Task 1**

   (a) **Enter your name and student ID** to the beginning of COMP108W08.java.

   (b) As stated above in 2a, we want to look at each movie ID in the requests and report if it does not exist in the database. We can do this by nested loops: the outer loop iterates through the requests one by one and the inner loop iterates through the database one by one and report the movie ID if it does not exist in the database.

   (c) Implement the method in **notExists()** of COMP108W08.java. This method takes 2 parameters: the first is the array storing the requests and the second is the size of this array. The database is stored in the list with the first node pointed to by `head` and last note pointed to by `tail`.

   (d) What is the time complexity of your algorithm? Justify your answer. Give your answer in the comment section at the beginning of COMP108W08.java.

4. **Programming with Nested Loops Task 2**

   (a) This time we would like to go through each element of the database and **count** how many times it appears in the requests. We also do this using nested loops.

   (b) Implement the method in **count()** of COMP108W08.java. The order of the 2 parameters is the same as Task 1.

   (c) What is the time complexity of your algorithm? Justify your answer. Give your answer in the comment section at the beginning of COMP108W08.java.

5. Testing data: You can try the following input

   The database contains eight entries: 70 20 60 40 50 30 10 80

   The requests contains twenty entries 5 10 60 70 15 50 30 20 20 20 25 15 20 10 20 5 70 70 10 10

   The expected outcome for notExists() is 5 15 25 15 5 and the expected outcome for count() is 3 5 1 0 1 1 4 0