**Math Question Generator with different data structures (Hashset, Linkedlist, Arraylist)**

**Final Project Documentation**

A logo for a university

AI-generated content may be incorrect.

Kaneko Kenichi Josiah (2902647462)

Jeremy Nathanael Gunawan (2802522960)

**Class L2BC**

**Computer Science Program**

**School of Computing and Creative Arts**

**Jude Joseph Lamug Martinez MCS**

**Dr. MARIA SERAPHINA ASTRIANI, S.Kom., M.T.I.**

**Data Structures**

**Object-Oriented Programming**

**Bina Nusantara International University**

**Jakarta**

TABLEOFCONTENTS A.BACKGROUND..................................................................................................................1 B.PROBLEMDESCRIPTION..............................................................................................2 C.PROPOSEDSOLUTION...................................................................................................3 1.ArrayList..........................................................................................................................3 2.LinkedList.......................................................................................................................3 3.Hashset..........................................................................................................................4 7.Results..............................................................................................................................7 Time.............................................................................................................................18 Space............................................................................................................................22 8.ImplementationoftheBestOneinApplication.............................................................23 D.TEAMWORKLOADANDTHINGSTODO...............................................................23 REFERENCES.......................................................................................................................24 APPENDIX.............................................................................................................................25

Background:

In these modern times, mathematics has become a pillar to critical thinking, problem-solving, and logical reasoning skills have continued to increase. However, many have failed to realize that the simplest arithmetic mathematics problems are still as important as a daily necessity. Traditional teaching methods often rely on static problem sets that rely on a strong basis on its roots; however, that is a problem when. It is shown that students with variability in practice while also experiencing similar but progressively difficult problems have enhanced their retention and adaptability. This underscores the need for accessible tools such as randomized math problem generators- creating endless questions that one can customize the difficulty.

In terms of experience and from others as well, they have experienced a multitude of problems even if the tasks were only arithmetic operations. Some have said that, even though they have finished their exam an hour early, they experienced a lot of careless calculations as the exam does not tolerate calculators. I may even be included to that sort of situation as my marks were affected just because of careless mistakes in simple arithmetic operations. It is not that one could not concentrate that makes these kinds of situations appear, but there are a variety of factors that induce this phenomenon. Such factors mentioned include: the stress induced by exams, the barrage of calculations needed without a calculator that creates misreading, transfer and encoding errors, and miscalculations.

Creating this kind of program (math question generator) also tackles computational efficiency with educational effectiveness. The system we have developed is programmed in hopes to generate questions rapidly, uniqueness, adapt difficulty, and minimize memory overhead. Generating limitless questions is one thing, but being able to optimize the difficulties of the questions (e.g. increasing number of operands, digits) will aid in providing uniqueness that will prevent repetition.

This project also will tackle evaluating which data structures works best for a given program in circumstances. The data structures to be evaluated include: ArrayList, LinkedList, and Hashset. Furthermore, data structures is extremely important for effective programs to work efficiently in terms of space and time. Being adept to which data structures work best will be the building blocks of effective computer programs. Therefore, finding out and applying the best data structures for this random math question generator is of utmost importance to increase the effectiveness and efficiency of the program, resulting in the perfect math training program.

Problem description

Similarly to the mentioned importance of choosing the right data structure for the most effective program, the right data structure for a program will help various devices and it will ensure that those devices will experience the best performance. This math question generator allows thousands of questions which will be endless, and that more than 7 digits and operands is already a lot to deal with. Inefficient data structures will result the math question generator to be: slow question generation (increasing the digits and operands will take a lot of time especially when asking to generate more than 50 questions), high memory usage that will prevent the program from working and it will display an error as it will have difficulties storing thousands of questions, redundant questions (duplicate questions), and a non-smooth program will result in poor experience.

In this project, we will test 3 different data structures: ArrayList, LinkedList, and Hashset. Each data structure will be compared to one another to obtain data and provide results which data structure is best and their best-performing specific action (e.g. ArrayList possibly being the fastest in terms of generating a lot of questions in terms of digits and operands). We will obtain datas in terms of the speed and memory usage of: handling questions with higher amount of digits and operands, efficiency of generating higher number of questions, and which data structure is best for generating specific operations (addition, subtraction, division, multiplication, and mixed).

Calculations with the help of a specific program created for these programs will help comparing and obtaining an average for runtime speed and memory usage for each function of each data structure. The data structure with the best speed and memory usage will be considered the best data structure for this math question generation program.

Demonstration

Demosntration#0:

A screenshot of a computer program

AI-generated content may be incorrect.

There are 3 files that each function as the random math generator but with different data structures: Hashset, LinkedList, and Arraylist. The MathGen\_Performance\_Time\_Tester is optional to be used, the main function (though self-explanatory) is to provide the time-complexity and space-complexity of the different data structures.

Firstly, upon executing any of the 3 files (the math question generator programs), it will open a popup window like below (Demonstration#1). However, the program is supposed to be utilized in full screen (Demonstration#2).

Demonstration#1:

A screenshot of a computer

AI-generated content may be incorrect.

Demonstration#2:

A screenshot of a computer

AI-generated content may be incorrect.

Demonstration#3:

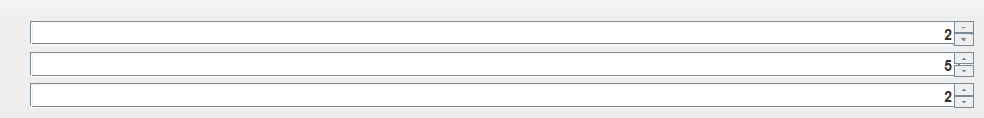
A screenshot of a computer

AI-generated content may be incorrect.

Once set to full screen, there are various functionalities and features that you will be able to see and experience. From the Demonstration#3, shows the number of functionalities the program has that all 3 files share.

Demonstration#4:

A close up of words

AI-generated content may be incorrect. 

The above is the first functionality. Users will be able to customize the number of digits, operands, and number of questions to be generated. To clarify, the number of digits means the number of digits per between operand- highlighted red. The number of operands mean the number of values with an operator symbol to be generated in each question- highlighted blue. Finally, the number of questions generated- highlighted green. In here, the number of questions, digits, and operand are all 5.

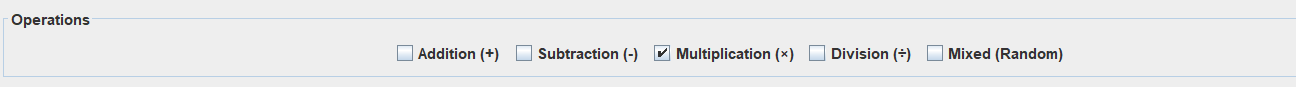
Demonstration#5:

A white square with black numbers and green circle

AI-generated content may be incorrect.A screenshot of a white board

AI-generated content may be incorrect.

Demonstration#6:



Here, is the second functionality. This requires users to provide input on which kind of operation for the questions that they would like to answer or have the questions generated. The functionality only works with only one input, meaning that even if the program allows multiple inputs, only one input will be taken. For example, ticking both multiplication and division means that recent tick will be submitted. As a heads up, the mixed category means that for each question, they will have only one focused operation symbol but the next question will be different or randomized. To summarize, the program only allows one operation input with mixed having each question having a randomized operation but in one question it will have the same operation. In Demonstration#7, this is a mixed operation generated question with 5 operand, 5 digits, and 5 questions.

Demonstration#7:

A screenshot of a computer

AI-generated content may be incorrect.

Demonstration#8:

A screenshot of a computer

AI-generated content may be incorrect.

Demonstration#8 is the third functionality of the program. It is a simple button that starts generating the questions once and only if functionalities 1 and 2 has a user input already.

Demonstration#9:

A screenshot of a computer

AI-generated content may be incorrect.

In the image above, 3 new functionalities are displayed, this includes: Next question and your answer, the column box at the left, and finally the space where it indicates which question is being focused and answered right now.

A screenshot of a computer

AI-generated content may be incorrect.Demonstration#10:

A screenshot of a computer code

AI-generated content may be incorrect.

In the left box, users could actually click which question to focus on. Simply clicking it once will change which question is being focused on right now.



In the bottom side, users could input their answers in the empty box and clicking on “Next Question” will submit their answer for that specific question. If the user did not click “Next Question” on the final question, it will continue to the next question. However, clicking the Next Question function on the last question, will submit all their work. It will also open a popup window that will show each question, the correct answer to the question, the user’s answer, and showing whether it is correct or not as shown in Demonstration#12.

Demonstration#12:

A screenshot of a computer

AI-generated content may be incorrect.

In addition, when a user inputs an answer to a question and goes Next Question (but not at the final question), the user can click onto the question they would want to check again and to see their answer. Furthermore, they could also erase and input a new answer in the box and then resubmitting their renewed answer by clicking “Next Question” again. This is demonstrated below with a detailed explanation on each picture.

Demonstration#13: Shows user inputting their answer “23” and submitting by clicking Next Question.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.Demonstration#14: Shows what happens after clicking Next Question- which is progressing to the next question.

A screenshot of a computer

AI-generated content may be incorrect.

Demonstration#15: Shows user going back to the previous question by clicking once onto that question. Which at first, shows “23”. User then renewed their answer by inputting their new answer and then resubmitting by clicking “Next Question”.

A screenshot of a computer

AI-generated content may be incorrect.

Demonstration#16: Shows the results and that user resubmission of their answer has been updated.

A screenshot of a computer

AI-generated content may be incorrect.

Program functionality has all been covered for the random math question generator. To summarize, executing the program displays a popup window. There, users can customize the difficulty by inputting their desired amount of digits and operands, and also choosing how many questions to generate by the program. Questions will be displayed with the current question focused displayed in the middle while the other questions will be displayed at the left column box. Users can input their answers by clicking on the empty box at the bottom and then input their answers with clicking on the “Next Question” button to submit their answers. Likewise, users can return to previous or go to other questions by clicking once on the questions at the left column box where they are able to view their answer for that question in the box and even resubmitting by reinputting their answer and then clicking “Next Question” again. Clicking “Next Question” on the last question will display another popup window that shows all user’s answers, questions, the correct answers, and displaying feedback whether user’s answers are correct or not.

Video link demonstration of program:

References:

Think Academy. (2023). Avoiding careless mistakes in math exams: a parent’s guide.

<https://www.thethinkacademy.com/blog/avoiding-careless-mistakes-in-math-exams-a-parents-guide/#:~:text=Understanding%20Careless%20Errors%20in%20Math,subtraction%2C%20multiplication%2C%20and%20division>.

Appendix:

Github repo (source code, report, PPT, documentation, demonstration):

<https://github.com/kanekojosiah/aaaaa.git>

[GitHub - kanekojosiah/aaaaa](https://github.com/kanekojosiah/aaaaa)