Formal Methods of Identifying Computer Science Problems

Student's Name

Institution

Formal Methods of Identifying Computer Science Problems

Three methods that help to identify computer science problems

Computer science refers to the study of solutions, problem-solving, and problems that are associated with the problem-solving process. Ideally, a computer scientist plays the role of developing an algorithm that comprises step by step instructions that helps solve any occurrence of a likely problem (Miller & Ranum, 2013). Therefore, computer science could be perceived as the study for algorithms. Nonetheless, it is important to acknowledge that some problems could not have a solution. It is true to say that computer science is about the study of problems without solutions and the study of solutions to problems. A user is required to have clear knowledge for the problem that needs to be solved to be able to instruct the computer appropriately.

Several methods could help identify the computer science problems. Some of these methods include abstraction that enables one to identify the problem and even the solution in such a manner that it can isolate the so-called physical and logical perspectives. The algorithm is another method that also helps in identifying a computer science problem through a step by step instruction that eventually helps to solve the problem that has been identified (Miller & Ranum, 2013). The algorithms are thought as finite procedures that help to solve a problem when all the steps have been followed. Fraction class is another method that could help identify the computer science problem through python paradigm.

Expansion to understand the selected method

According to Futschek (2006), algorithmic is essential in informatics that is often established freely from learning programming. The algorithmic method helps to utilize the problems that could not be easily solved but could be easily identified. The method does not help

in identifying the computer science problem but also helps in solving such problems. In essence, the algorithmic method comprises a pool of skills that are associated with the understanding and development of algorithms. It becomes easier to identify, analyze, and specify a problem. The major concern involves the ability of the method to identify problem including those that could not be solved. Therefore, the language used in establishing the algorithm needs to be problem-oriented and high-level.

Ways to utilize the results of the research in the course project

I will utilize the results in the course project to establish the way scholars conceive the important idea of efficiency. Besides, the results will help devise ways to implement the program. Notably, the design for an efficient algorithm will play a great role in identifying the problem which I deliberate to be among the most significant research areas in computer science. The study for algorithms will offer the insights especially to the problems included in offering methods for solutions that are independent of the implementation aspects or programming languages (Gal-Ezer & Zur, 2004). The results of the algorithmic method will help compare some algorithms that could solve it regarding time and space.

Ways to use the results of the research in the course project

The research findings will be used in some ways. The results could be used to acknowledge the impact that studies on misconception in light of the science and mathematical educations leading to the teaching of the algorithmic method. Besides, the results will help expose the misconceptions relative to the notion of efficiency based on its importance to the analysis of the computer science program. The results associated with the implementation of the program including interviews with the respondents will confirm the presumption on the

efficiency to grasp the concept of the algorithmic method during problem identification (Gal-

Ezer & Zur, 2004). The results will help identify ways to solve the problems and where possible

prevent them from reoccurring.

References

Futschek, G. (2006). *Algorithmic Thinking: The Key for Understanding Computer Science*

[Ebook] (pp. 159-168). Springer. Retrieved from

https://publik.tuwien.ac.at/files/PubDat_140308.pdf

Gal-Ezer, J., & Zur, E. (2004). The efficiency of algorithms—misconceptions. *Computers &*

Education, *42*(3), 215-226.

Miller, B., & Ranum, D. (2013). *Problem-Solving with Algorithms and Data Structures* [Ebook].

Retrieved from

https://www.cs.auckland.ac.nz/courses/compsci105s1c/resources/ProblemSolvingwithAlgor

ithmsandDataStructures.pdf