

Effective Learning on Efficiency of Algorithms Through Puzzle Based Learning

Rashmi K B

Department of ISE
BMS College of Engineering
Bangalore, India

Yashaswini H M

Department of ISE
Nitte Meenakshi Institute of Technology
Bangalore, India

Dr. Roopa R

Department of ISE
BMS College of Engineering
Bangalore, India

Abstract—In the present scenario of an extremely competitive world, there is a need of proficient graduates who are able to deal with real world problems. But traditional method of teaching was lagged to address this issue as students learning were limited to known textbook concepts and solutions. Hence outcome based education comes into picture which makes the students more knowledgeable, skillful and capable of facing real world scenarios by evaluating the students based on specific outcomes. Teaching and learning process in outcome based education is student centered learning. We have adopted one such teaching learning method for the course analysis and design of algorithms. The course Analysis and design of algorithms are based on the algorithms with respect to different design techniques. These different design techniques play a vital role in designing and analyzing efficient algorithms in terms of time and space complexities. To prepare the students, to apply and analyze the design techniques for real world applications we inculcated the puzzle based learning environment as a teaching learning process. In this paper, we discuss one of the implemented puzzle and evaluation rubrics which led the students to enhance the cognitive skills and is been measured using indirect survey.

Keywords—algorithms, analysis, efficiency, puzzle based, design techniques, rubrics.

I. INTRODUCTION

Learning is gaining some knowledge or information by several ways like studying, experiencing or taught by teachers. Outcome based education aims at lifelong learning so that students learning should not be limited to classrooms. Continuous learning is very crucial as it gives the ability to solve any real-world problems even after graduating from the college. There are several ways educators can teach the students. Earlier it was more likely to be chalk and talk where students were passive listeners. In this competitive world with the advanced technology teaching learning process is no more limited to chalk and talk. In this regard Outcome based education has been implemented where students are as active as educators. It includes puzzle based learning, project based learning, e learning, innovative methods [1].

Algorithms play a vital role in computer science domain. The algorithm is a set of statements is like a rough sketch before implementing the solution. Analysis and design of algorithms involve designing an efficient algorithm by minimizing time and space complexities [2]. The Traditional way of teaching was theoretical, students thinking were limited to textbook based algorithms. Puzzle based learning involves students in understanding, analyzing and solving the given puzzle in terms of algorithmic implementation. students are able to implement the real-world scenarios [3,4]. In a class, each team of two was given a puzzle. They were involved in understanding the puzzle implement the solution through design algorithms. This enhances the critical thinking and interest about algorithms.

II. PUZZLE BASED APPROACH

According to Zbigniew Michalewicz Matthew Michalewicz, traditional teaching learning process was not concentrating on problem-solving skills. Students were tended to learn what is prescribed and known concepts which made their knowledge was limited to exams. When students encountering real world problems without any instructions to follow, they found it difficult to analyze to find the solution [5].

Introducing puzzle based learning in the course makes the students think creatively and in a blissful manner. Puzzle builds enthusiasm, worthiness, and relevance to the subject. This leads student's involvement and provoking thoughts. Puzzle based learning enhances thinking skills, problem-solving skills in a technical approach [6,7].

Companies in need of skilled graduates with good thinking and analytical skills. Therefore, we have to mend students in such a way that they should have the ability to apply their knowledge to solve any technical issues [8].

As Fisher (2001) wrote: “. . . though many teachers would claim to teach their students ‘how to think’, most would say that they do this indirectly or implicitly in the course of teaching the content which belongs to their special subject. Increasingly, educators have come to doubt the effectiveness of teaching ‘thinking skills’ in this way, because most students simply do not pick up the thinking skills in question” [9].

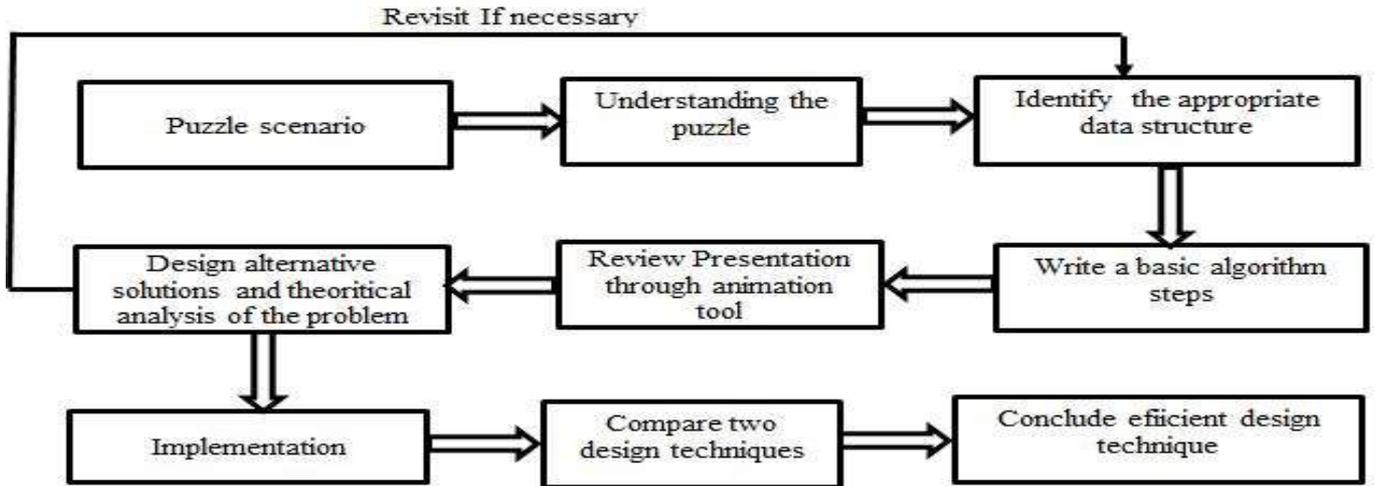


Fig. 1. Phases in Puzzle Based Learning

The course analysis and design of algorithms is a comprehensive course along with self-study, different puzzles are given to each team of two students. Students made to understand the puzzle, design and analyze an algorithm by the right choice of design methodology to implement the solution. Three reviews are conducted to evaluate their work based on rubrics as shown in the Fig. 1. One such puzzle is illustrated as a reference which highlights the work done by the students.

PUZZLE: Given a collection of n bolts of different widths and n corresponding nuts. We are allowed to try a nut and bolt together, from which we can determine whether the nut is larger than the bolt, smaller than the bolt, or matches the bolt exactly. However, there is no way to compare two nuts together or two bolts together. The problem is to match each bolt to its nut. We have to design an algorithm for this problem with average-case efficiency in (nlogn).

Review 1: Ability to analyze the problem and demonstrate the design solution. The details of rubrics and outcomes are shown in Table I.

Table I. Review1- Rubrics and Outcomes

Sl.no	Rubrics	Outcomes
1	Able to provide Logical interpretation of the problem.	Pick a nut and check it with each bolt for matching Check whether the bolt is larger, smaller or of the same
2	Identifies appropriate data structures using prerequisite knowledge	Array is identified as the appropriate data structure
3	Concise basic algorithmic steps based on logical interpretation	<ul style="list-style-type: none"> • If the bolt is smaller, put it into new array: small • If it is larger than the nut, put it into another new array: large • If it matches with the nut store the size of nut into a variable • Pick a new nut, compare it with the updated variable • If it is smaller search in the new array • small, else search in the array large • Repeat all the steps recursively till all the nuts are matched with their corresponding bolts.
4	Use of animation tool for presentation	PowToon Animation software used to present review 1

Outcome of Review 1: Students were able to understand and analyze real world problem in terms of algorithm designing. This review aimed to enhance their communication skills, team work, and modern tool usage.

Review 2: Report/document that includes the theoretical analysis of the problem. The details of rubrics and outcomes are shown in Table II.

Table II. Review2- Rubrics and Outcomes

Sl No.	Rubrics	Outcomes
1	Able to Recognize the appropriate design strategy for the given problem	1. Divide and Conquer 2. Brute Force
2	Justification of the design strategy opted by comparison with any other design techniques	Used divide and conquer technique so that we will have Better average time efficiency Compared to any other technique that can be used.
3	Algorithm based on the justification provided	Algorithm is given below
4	Theoretical Analysis of best, average and worst- case inputs for the algorithm provided	BEST CASE: The best case is when nuts and bolts are arranged in same order. AVERAGE CASE: The average case is when bolts are randomly arranged WORST CASE: The worst case when bolts are arranged in reverse order to the order in which nuts are arranged
5	Theoretical analysis of their complexity	For Divide and Conquer: BEST CASE: $\Omega(N \log N)$ WORST CASE: $\log N$ AVERAGE CASE: $\Theta(N \log N)$ For Brute Force BEST CASE: $\Omega(N)$ WORST CASE: AVERAGE CASE: $\Theta(N^2/2)$

Outcome of Review 2: Students were able to recognize different design techniques, decide appropriate design technique for the given puzzle. It enhances algorithm writing, report writing abilities. They will get an idea about time complexities based on average, worst and best cases.

1.Divide and Conquer: In this view, students take first nut as reference nut to compare with the bolts, if the size of bolt is less than size of reference nut then we will put that bolt into an array (i.e. small array) otherwise put that bolt into another array (i.e. large array) and do it recursively until all nuts are compared with all bolts.

1. Input: Two arrays (i.e. nut[] and bolt[])
2. Output: To match the given nuts to bolts
3. Select a nut
4. for $i \leftarrow 0$ to n (compare bolt[i] to nut)
5. If bolt is smaller than nut we will put into small[] array
6. If it is larger we will put into large array
7. If it is equal print match found
8. select next nut from the nut array
9. If it is smaller than the previous, search in the small array.
10. If it is larger than the previous, search in the large array.
11. Do recursively until every nut is matched with its corresponding bolt

2. Brute Force: In this method, students selected a nut from the given nut array (position i) and compare each bolt with selected nut using sequential search, when an element in the bolt matches with the selected nut (position j) they swap the matching bolt with the bolt which is in the i^{th} position.

1. Input: Two arrays (i.e. nut[] and bolt[])
2. Output: To match the given nuts and bolts
3. Select a nut
4. for $i \leftarrow 0$ to n (select a nut)
5. for $j \leftarrow i$ to n (select a bolt and compare with nut)
6. If $bolt[j] = nut[i]$ swap($bolt[i], bolt[j]$) and print match found

Review 3: Implementation of the algorithm, Able to Analyze the time efficiency. Review3 rubrics is as follows.

1. Able to provide Executable programming solution using the alternative design strategies specified in the documentation.
2. Validation for best, average and worst-case inputs
3. Valid results for randomized inputs.

- 4. Use of advanced programming language for coding.
- 5. Analyze the time complexities for the two alternative design solutions implemented.
- 6. Based on analyzed time complexities provides the valid conclusions for the efficient design technique.

Outcome of Review 3: Students were able to code for those algorithms using advanced programming language, executed and shown the output for both design strategies.

Example: Consider a nut array of 10 nuts. $Nut [] = \{2,4,6,8,1,5,11,14,9,7\}$

Best case: bolts[]={2,4,6,8,1,5,11,14,9,7} **Average case:** bolts[]={14,9,6,11,1,5,8,2,4,7} **Worst cases:** bolts[]={7,9,14,11,5,1,8,6,4,,2}

Time analysis for divide and conquer approach:

There are two arrays of Nuts and Bolts. For the selected nut, if the bolt is smaller than the nut, put that bolt in small array otherwise put that bolt in a large array. At each step, an array will be split into two halves. Therefore, in an average case, it reduces the complexity by a factor of 2 as shown in Fig. 2. In this scenario, time complexity will be $O(n \log n)$.

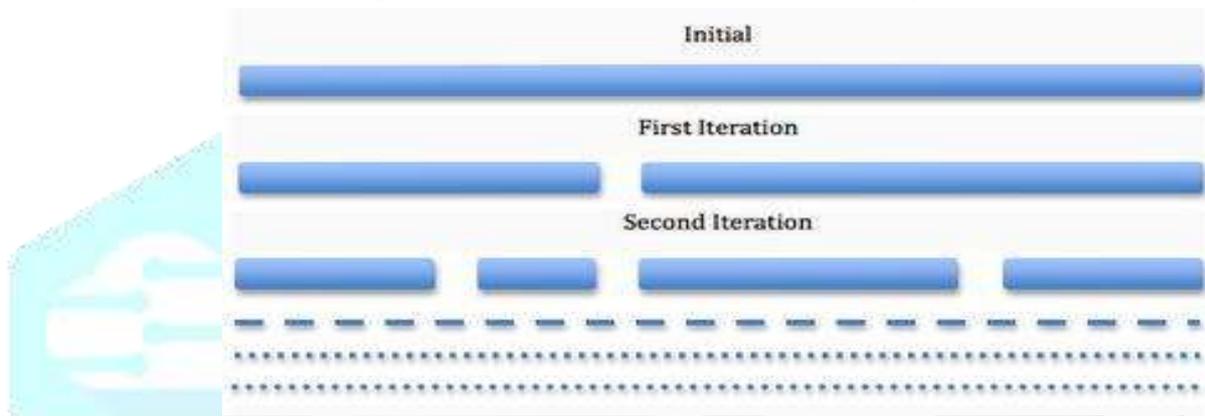


Fig. 2. Splitting Process of Divide and Conquer Approach

In the worst case, the selected nut or bolt may be the smallest or largest, in that case, one array will not be having any nut or bolt, other array will be having all the nuts or bolts. In this scenario, time complexity will be $O(n^2)$. The comparison between average and worst-case performance for divide and conquer approach is shown in Fig. 3.



Fig. 3. Performance Graph - Divide and Conquer Approach

The third review was evaluated by the external entity. It improves students coding skills and communication skills. Students will learn to maintain coding standards.

III. INDIRECT SURVEY

The course outcomes were measured through course end survey questionnaires as in table III below which are framed based on various levels of bloom’s taxonomy such as remember/understand, apply, analyse and design.

Table III: Course End Survey questionnaires

Sl.No:	Questionnaires					
		Excellent	Very Good	Good	Fair	Poor
1	Were you able to understand the fundamentals of algorithm					
2	Were you able to identify the order of magnitude of an					
3	Were you able to understand and apply various desing					
4	Were you able to analyze the efficiency and correctness for					
5	Were you able to compare algorithms based on their desing					
6	Were you able to desing efficeint algorithms in common					
7	Were you able to create an efficient algorithm using					
8	Rate your proficieny in learning the modern tool for					

The attainment level of the above questionnaires are as in table IV:

Weightage								
5(Excellent)	23	20	23	21	20	23	23	19
4(Very Good)	22	23	23	20	25	17	20	20
3(Good)	7	7	5	9	5	10	8	11
2(Fair)	0	2	0	2	1	1	0	1
1(Poor)	0	0	1	0	1	1	1	1
Total no of students	52	52	52	52	52	52	52	52
Attainment level	0.86	0.8	0.86	0.8	0.8	0.8	0.8	0.8
In %	86.2	83	85.8	83	84	83	85	81
10% weightage	8.62	8.3	8.58	8.3	8.4	8.3	8.5	8.1

CONCLUSION

This paper inscribes how the students could be prepared technically to learn and implement the challenging problems with the different way of teaching methodology that is puzzle based learning. Each team in the class explored the design techniques and could understand, apply, analyze and implement the puzzles assigned to them with two different design techniques. The indirect survey measures outcome is that through Puzzle based learning students could apply, analyse and design an efficient algorithm with the valid conclusion in terms of time efficiency and also could enhanced their knowledge and skills.

REFERENCES

- [1] Felder, Richard M., and Linda K. Silverman. "Learning and teaching styles in engineering education." *Engineering education* 78.7 (1988): 674-681
- [2] Naragund, Jayalakshmi G., and Vidya S. Handur. "Educationally effective teaching of design and analysis of algorithms." *Innovation and Technology in Education (MITE), 2013 IEEE International Conference in MOOC*. IEEE, 2013.
- [3] Xuemin Zhang, Zenggang Xiong, Yaomin Ding, Guangwei Wang & Conghuan Ye. "Research-oriented teaching of an Algorithm Design and Analysis course." *World Transactions on Engineering and Technology Education* Vol.12, No.3, 2014: 534-538.
- [4] Merrick, Kathryn E. "An empirical evaluation of puzzle-based learning as an interest approach for teaching introductory computer science." *IEEE Transactions on Education* 53.4 (2010): 677-680.
- [5] Michalewicz, Zbigniew, Nickolas Falkner, and R. Sooriamurthi. "Puzzle-based learning: An introduction to critical thinking, mathematics and problem solving." *decision line* 42.5 (2011): 6-9.
- [6] Falkner, Nickolas, Raja Sooriamurthi, and Zbigniew Michalewicz. "Teaching puzzle-based learning: development of transferable skills." *Teaching Mathematics and Computer Science* 10.2 (2012): 245-268.
- [7] Falkner, Nickolas, Raja Sooriamurthi, and Zbigniew Michalewicz. "puzzle based learning for engineering and computer science" *computer* 43.4(2010):20-28
- [8] Falkner, Nickolas, Raja Sooriamurthi, and Zbigniew Michalewicz. "Teaching puzzle-based learning: development of basic concepts." *Teaching Mathematics and Computer Science* 10.1 (2012): 183-204.
- [9] A. Fisher, *Critical Thinking: An Introduction*, Cambridge University Press, Cam-bridge, UK, 2001