# A Worked Example Model for Teaching Dynamic Programming

Ed Novak
enovak@fandm.edu
Franklin and Marshall College
Lancaster, PA, USA

## ABSTRACT

How should dynamic programming be taught to students experiencing it for the first time? Dynamic programming is a sophisticated programming technique that exercises many aspects of computer science in concert. Because of the deep technical complexity therein, building effective lessons is challenging. In this work we propose a worked example model for teaching dynamic programming that centers around a midterm exam in which the solutions are provided to the students weeks in advance. 35 students were surveyed about their experiences learning dynamic programming with and without this model.

## 1 INTRODUCTION

Dynamic programming is an advanced programming technique that achieves substantial efficiency gains over straightforward recursion for certain problems. Teaching dynamic programming to those that are first experiencing it is challenging due to the sophisticated nature of the problems and solutions. Exams that expect students to solve new (to them) dynamic programming problems are unfair given a limited amount of time (~80 min) and experience.

In this work, we propose a "worked example" methodology for teaching dynamic programming that leverages several important pedagogical techniques such as learning by testing one's self, self-paced learning, meta-cognition, content interleaving, and Bloom's taxonomy [3]. Our methodology is centered around a special midterm exam in which the students are given the solutions (a worked example) in advance.

## 2 BACKGROUND

Educating students using worked examples is not a new concept. But, our use of it is novel in teaching dynamic programming. A foundational work in the field of worked examples is that of Atkinson et al. [2]. Worked examples are common in many educational contexts (especially math), but it is less commonly studied in higher level courses within computer science. Two notable works are [5] and [4], which both are exemplary of the most common idea: providing code examples to novice programmers. The recent work from Akhuseyinoglu et al. [1] demonstrates that worked examples can be useful outside of the CS1 context.


*SIGCSE 2023, March 15–18, 2023, Toronto, ON, Canada*

ACM ISBN 978-1-4503-9433-8/23/03.
https://doi.org/10.1145/3545947.3576232

## 3 OVERVIEW OF METHODS

In the proposed teaching model, an elaborate worked example of a dynamic programming problem and solution is given to students. The solution is broken into several parts including working the problem by hand, providing a recursive solution, showing the repeated work in the recursive solution, and providing a dynamic programming solution. After weeks of studying the worked example via in-class activities and assignments, a midterm exam is given containing the same problem. Students are expected to thoroughly re-create the solutions from the worked example from memory.

In the Summer of 2022, 35 students were surveyed after having taken the same course (CPS-222) at various points in the past. 14 participants experienced the worked example model and 21 took the course before the introduction of the worked example model. Survey questions were written along themes of confidence, test anxiety, and general relevancy of dynamic programming.

## 4 CONTRIBUTIONS AND FUTURE WORK

Our key findings are that students who experienced the worked example model felt less stress and less anxiety regarding the midterm exam. Students also expressed the notion that the relevancy of dynamic programming is artificially boosted by academia and so called "technical interview" questions.

In the future, we plan to examine whether or not the worked example model improves students' understanding and competence when solving dynamic programming problems.

## REFERENCES

[1] Kamil Akhuseyinoglu, Ryan Hardt, Jordan Barria-Pineda, Peter Brusilovsky, Kerttu Pollari-Malmi, Teemu Sirkiä, and Lauri Malmi. 2022. A Study of Worked Examples for SQL Programming. In *Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 1* (Dublin, Ireland) *(ITiCSE '22)*. Association for Computing Machinery, New York, NY, USA, 82–88. https://doi.org/10.1145/3502718.3524813

[2] Robert Atkinson, Sharon Derry, Alexander Renkl, and Donald Wortham. 2000. Learning from Examples: Instructional Principles from the Worked Examples Research. *Review of Educational Research* 70 (06 2000), 181. https://doi.org/10.3102/00346543070002181

[3] Elizabeth Bjork and Robert Bjork. 2011. Making things hard on yourself, but in a good way: Creating desirable difficulties to enhance learning. *Psychology and the Real World: Essays Illustrating Fundamental Contributions to Society* (01 2011), 56–64.

[4] Wengran Wang, Audrey Le Meur, Mahesh Bobbadi, Bita Akram, Tiffany Barnes, Chris Martens, and Thomas Price. 2022. Exploring Design Choices to Support Novices' Example Use During Creative Open-Ended Programming. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 1* (Providence, RI, USA) *(SIGCSE 2022)*. Association for Computing Machinery, New York, NY, USA, 619–625. https://doi.org/10.1145/3478431.3499374

[5] Rui Zhi, Thomas W. Price, Samiha Marwan, Alexandra Milliken, Tiffany Barnes, and Min Chi. 2019. Exploring the Impact of Worked Examples in a Novice Programming Environment. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (Minneapolis, MN, USA) *(SIGCSE '19)*. Association for Computing Machinery, New York, NY, USA, 98–104. https://doi.org/10.1145/3287324.3287385