A Worked Example Model for Teaching Dynamic Programming

Ed Novak, Franklin and Marshall College
SIGCSE TS 2023

ABSTRACT

How should dynamic programming be taught?

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.

INTRODUCTION

- Dynamic programming is an advanced programming technique that can achieve substantial efficiency gains for certain problems.
- Dynamic programming is sophisticated and the apt problems are very complex.
- Worked examples are commonly used in lower-level content (introductory CS sequence, math, etc.), but have not been explored thoroughly for upper-level advanced topics.
- Worked examples have synergy with many other well known pedagogical concepts such as self-testing, self-paced learning, meta-cognition, and Bloom's Taxonomy
- We believe worked examples can serve as a powerful framework for learning extremely complex and detailed topics such as dynamic programming.

Survey Details

- N = 35 undergraduate students studying computer science at a small, private liberal arts college.
- Students were recruited from undergraduate courses (CPS-222 Intro CS3, and CPS-261 Algorithms).
- Fall 2019 – present.
- Participation was incentive with $50 VISA gift cards.
- 77 were invited of which 35 chose to complete survey.

Name	Number
Control Group	21
Experimental Group	14
Total	35

DP Solution Process

1. Understand the problem
Be able to find a solution by hand (intuitively / brute force) for a simple example instance.

2. Identify problem is apt for DP solution
Demonstrate the greedy algorithm sometimes fails

3. Design recursive solution
   3a. Implement recursive solution in code
   3b. Analyze time complexity of recursive solution
   3c. Recognize repeated work in recursive solution

4. Design dynamic programming solution
   4a. Implement dp solution in code
   4b. Analyze time complexity of dp solution
   4c. Recognize that dp solution avoids repeated work

"Worked Example" Fragment

$[X_0, X_1, X_2, X_3, X_4, X_5, \ldots]$

Suppose newline is inserted here, the resulting solution may look like this:

```
[X0]          cost = (w-X0)^2
[X1, X2, X3] cost = (w-X1-X2-X3-1-1)^2
[?]
...
```

We call [x0] "left"
We call $[x_1, x_2, x_3, x_4, x_5, \ldots]$ "right"

We must try all different possible "newline" insertion points and use recursion to find the optimal solution. In general...

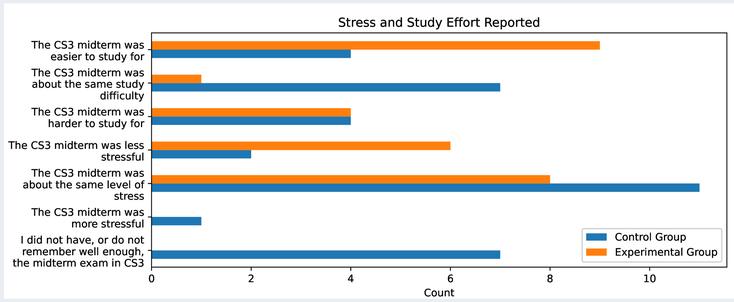
$$f(w, vec) = cost(left) + f(w, right)$$

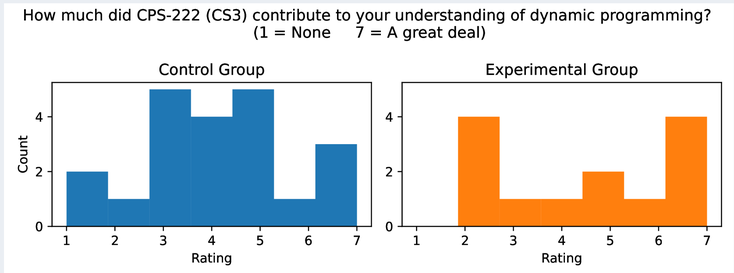
Specifically...

```
Ans1 = cost([X0]) + f(W, right)
Ans2 = cost([X0, X1]) + f(W, right)
Ans3 = cost([X0, X1, X2]) + f(W, right)
...
```

Note: The value of "right" is different in each of these calls

Results





"I used dynamic programming a little when practicing for job interviews and during job interviews, but it wasn't as important as I thought it would be [...]"
– Anonymous

REFERENCES

Kamil Akhuseyinoglu, et al. 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).

Robert Atkinson et al. 2000. Learning from Examples: Instructional Principles from the Worked Examples Research. Review of Educational Research 70 (06 2000).

Elizabeth Bjork et al. 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).

Wengran Wang et al. 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)

Rui Zhi et al. 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).