Name: Michael Rogers

ID: 105667404

CSCI 3104, Algorithms **Profs. Grochow & Layer**

Explain-It-Back 7 **Spring 2019, CU-Boulder**

Explain dynamic programming to a biology major—what it is, how it works, and why it is valuable.

Name: Michael Rogers
ID: 105667404

CSCI 3104, Algorithms **Profs. Grochow & Layer**
Explain-It-Back 7 **Spring 2019, CU-Boulder**

Dear fellow CU biology student,

I understand you would like to get a better understanding on how dynamic programming works. Wonderful! Dynamic programming is an incredibly innovative and useful piece of coding. I will do my best to explain what dynamic programming is without getting too technical, but I would like to explain a couple of concepts that are crucial to dynamic programming before I continue. Dynamic programming is used to optimize a certain kind of programming called recursion. Recursion is used to go through data by using a function in which it calls itself. I know that sounds confusing on the surface but let me elaborate through examples. Assuming you are familiar with some basic statistics, a great example is a factorial. If we look at a factorial: 5! = 5 * 4 * 3 * 2 * 1 we can see that each multiple in the equation is 1 less than the number before it. Let's think about this in terms of each multiple in 5! . If 5 is the number, we want to find the factorial we would start by multiplying 5 by 5 - 1. We can then take the result of that and multiply it by 5 - 2, and so on until we reach 5 - 4. This unravels the factorial until the program multiplies the result by 1 and you have your result. Another good example to think of is the Fibonacci sequence, a naturally occuring sequence that can be seen all over the world (In plants and animals alike). As you are a biology student I won't go into detail on that now, I'm sure it has come up in your studies. Now that we have discussed what recursion is, we can move on to dynamic programming and how it can be used to optimize recursion. With recursion, we are trying to solve one big problem. However, while solving this problem we run into sub-problems that also contribute to the solution. In recursion, sometimes a certain sub-problem can pop up multiple times, meaning we have to solve it multiple times which we all know would be a waste of time and computing power. It is very inefficient to recompute a problem instead of storing its solution to be accessed as the answer to another sub-problem more quickly. This can be accomplished with dynamic programming. This strategy makes the program much more efficient because we aren't solving the same progblem multiple times. Now that we have seen how dynamic programming works and what it is, we can look at its value. If we are running a program on an extremely large dataset using recursion, recomputing a problem can begin to greatly affect how fast the program is. However, with dynamic programming, we don't have to recompute problems, which in turn makes the program run significantly faster. Problems that could not be solved because of the waste of computing can be solved relatively quickly using dynamic programming. I hope this was helpful, if you need more elaboration on anything just let me know.

Good Luck,
A fellow CU computer science student