Advanced Algorithm HW3

Seungho Jang

September 2020

1 Q1

To answer this question, where dynamic programming is applicable: optimal substructure and overlapping sub-problems. One way to solve problem with dynamic programming is to solve and characterize the structure of an optimal solution. The meaning of this is basically the solution of entire problem includes solutions of sub-problems. Also, overlapping sub-problem is pretty simple concept. If we were to draw tree of Fibonacci number, we might have some sub-problems retrieve same outputs. Those sub-problems will be overlapped in other child nodes. One problem can be applicable for dynamical programming is **Unweighted Shortest Path**

This concepts and idea can lead to the conclusion that we should be careful of using dynamical programming if we don't have a sub-problems are overlapping and the solution of problem doesn't include the solution of sub-problem.

By contrary to **Unweighted Shortest Path**, **Unweighted longest simple path** does not applicable for dynamical programming because it doesn't only exhibit the optimal substructure, but also the solution from sub-problem won't necessarily be included for the problem.

2 Q2

If we were to think rod-cutting problem, the rod-cutting problem can be solved in recursive way by showing the maximum revenue.

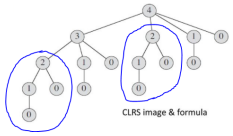

Figure 1: Recursive Tree - Rod-Cutting Problem.

In given rod-legnth = 4, we can create a recursive tree by knowing that the problem can be solved recursively. However, the problem of recursive solution is it takes forever to calculate if rod-length gets longer. Then, we know that there

are two sub-problems can be overlapped just like in figure, then what we want to do is to store all the result of sub-problem result consecutively while ignoring the same result. Then the maximum revenue stored in the array, then we can compare the elements and output the maximum revenue. Since these elements is the sub-problem solution, it means that the one of the solution could be the main problem's solution. However, this does not shows where to cut the rod, so bottom-up approach would be very efficient and visible for us to observe the result because it does not only skip overlapped sub-problem's solution, but also shows the index(where-to-cut) in which provides maximum revenue

3 Q3

All the results and discussion can be found in this code link : Bottom Up Approach

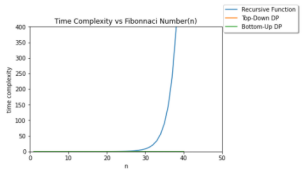

Figure 2: Recursive vs Top-down vs Bottom-up Approach Comparison.

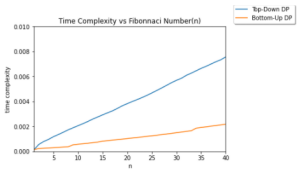

Figure 3: Top-down vs Bottom-up Approach Comparison.

The first plot above shows that recursive function of Fibonacci number is growing exponentially as compare to Top-down and Bottom up dynamic programming approach. Also, the second plot above shows that Top-down approach takes more time to calculate the Fibonacci number as compare to bottom-up approach. However, the assumption can be made that they are similar growth because they are in very small scale.

4 Q4

What is the matrix chain multiplication problem? Discuss how dynamic programming can be applied to solve it.

Matrix chain multiplication problem can be solved in recursively by considering total number of multiplication that we have to do with different dimension of matrix. For example, if we have to multiply all these matrices: 10*10, 10*5, 5*80, the total number of multiplication will be (10*10*5)+(10*10*80) as well as (10*5*80) + (10*10*80). Even though the result does not change, the order can be different. So then we can solve this recursively. Then using recursive definitions, it could meet one of the property of dynamic programming. Since the one of the different order could be the result of the entire problem solution, this also meet the other property of dynamic programming.

5 Q5

The code was ran and tested on this link : Longest Common Sub-sequence

1. Longest Common Sub-sequence between nsp2 and A0A1B3Q5V8 is AAQV with length of 4

2. Longest Common Sub-sequence between nsp2 and Q9YMB7 is QYM with length of 3

6 Q6

The code was ran and tested on this link : Longest Common Sub-sequence with the same function on question 5.

Longest Common Sub-sequence $\langle 1, 0, 0, 1, 0, 1, 0, 1 \rangle$ *and* $\langle 0, 1, 0, 1, 1, 0, 1, 1, 0 \rangle$ *is* $\langle 0, 1, 0, 1, 0, 1 \rangle$ *with length of* 6