

# WOA7001 Pre-test

**Name:** YangZhenyu

**Matric number:** s2118292

In order to better understand our project, I have made some understanding of optimization algorithms in pre-test and pro-test respectively.

Here is the pre-test: An understanding of dynamic programming optimization algorithms.

Dynamic Programming is mainly an optimization over plain recursion. Wherever we see a recursive solution that has repeated calls for same inputs, we can optimize it using Dynamic Programming. The idea is to simply store the results of subproblems, so that we do not have to re-compute them when needed later. This simple optimization reduces time complexities from exponential to polynomial.

It studies the algorithm method of optimization problems by splitting them into several simpler subproblems. It is worth noting that the global problem depends on the optimal solution of its sub-problems. Thus, a very important feature of DP is the proper construction of optimization problems into multiple levels that are addressed sequentially at a time. Each level is solved by using ordinary optimization problem techniques, and the solution helps define the characteristics of the next level of problems in the sequence. Typically, levels represent different time periods in the overall problem landscape.

For example, if we write simple recursive solution for Fibonacci Numbers, we get exponential time complexity and if we optimize it by storing solutions of subproblems, time complexity reduces to linear.

<pre>int fib(int n) {     if (n &lt;= 1)         return n;     return fib(n-1) + fib(n-2); }</pre>	<i>Recursion : Exponential</i>
<pre>f[0]=0 f[1]=1 for (i=2, i&lt;=n; i++) {     f[i] = f[i-1] + f[i-2]; } return f[n]</pre>	<i>Dynamic programming : linear</i>