

Dynamic Programming in Assembly Line

Lo Luca

S4121569

The problem described is the scheduling of two assembly line and each with a number of station where a single operation is performed to the product.

At first the problem is solved using a brute force approach where we compute all the solution possible and then, look for the optimal one and return it. This approach is made using a recursive function where at each call make a recursive call in the case if we want to keep using the same assembly line or switch to the other one.

The problem with this approach is the cost that increase exponentially and in fact the time complexity is $O(2^n)$ and the auxiliary space is $O(n)$

The problem behind the brute force is the presence of repetitive computation of the same value that increase with the number of n =number of station.

With dynamic programming, we store the new value in an array and retrieve the value when we needed. Also it transforms the problem in subproblems, computing the optimal solution of the subproblem.

If we consider our problem like a binary tree where the decision is “proceed with the current line” or “go to the other line”, the brute force approach, visit all the leaf node while the dynamic programming visit just a subset of the tree.

The time complexity of the dynamic programming approach is $O(n)$ because we compute the value only n times and the auxiliary space is $O(1)$ because we just need a function, not a recursive one.