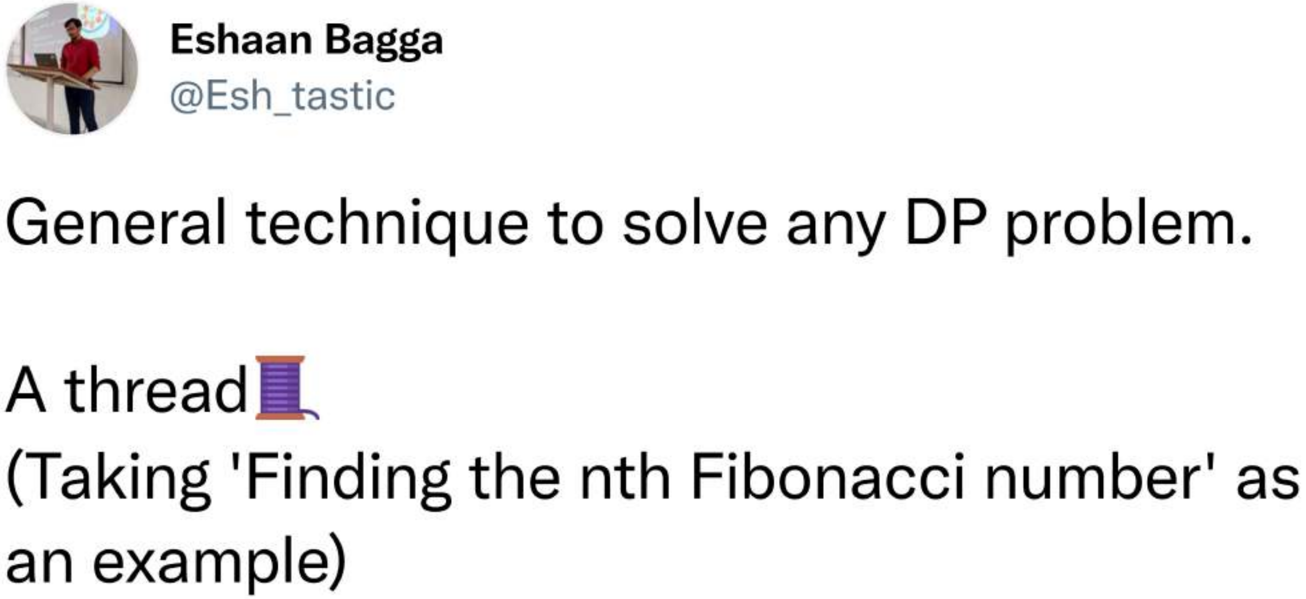
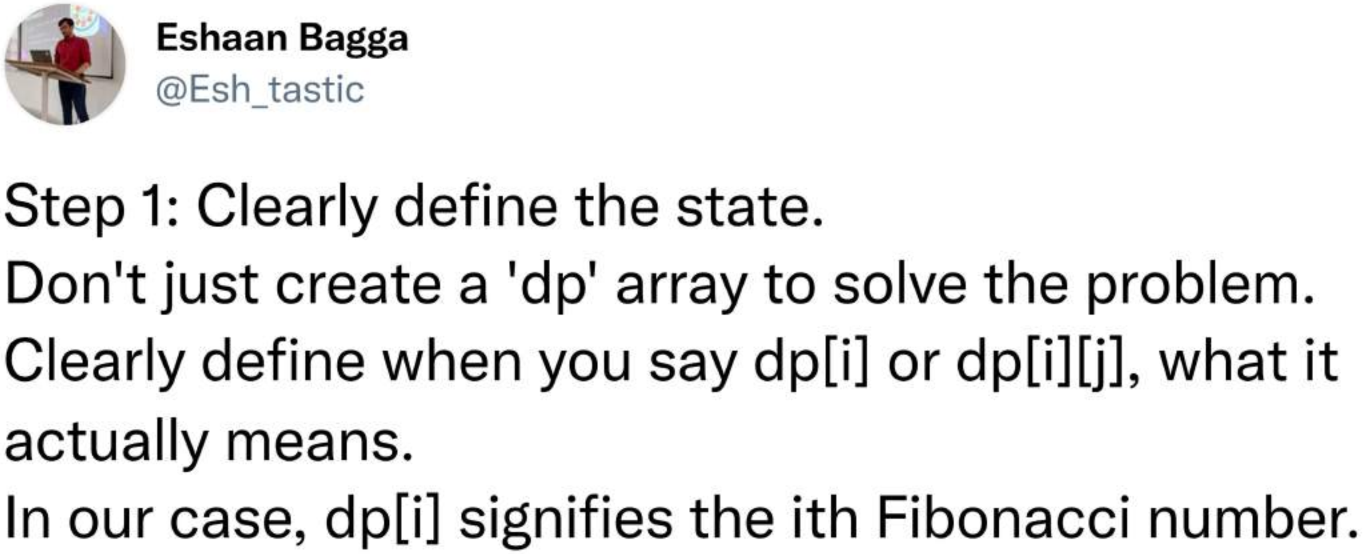


Eshaan Bagga
@Esh_tastic

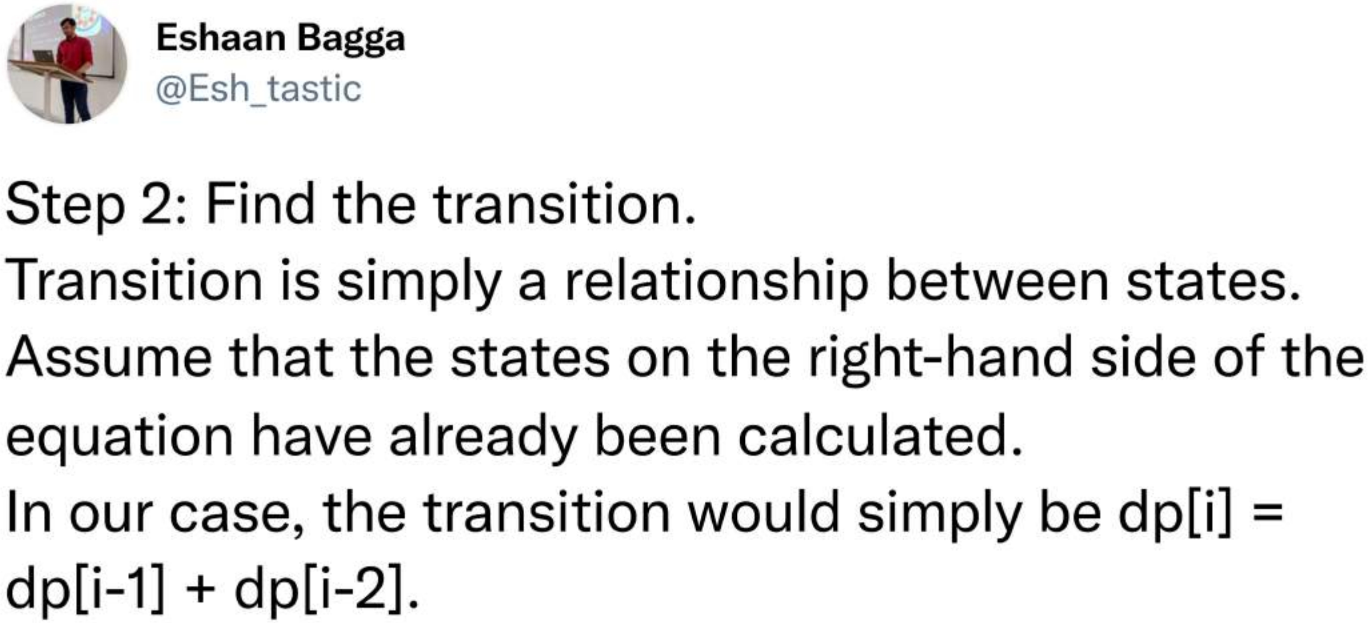
General technique to solve any DP problem.

A thread🧵
(Taking 'Finding the nth Fibonacci number' as an example)



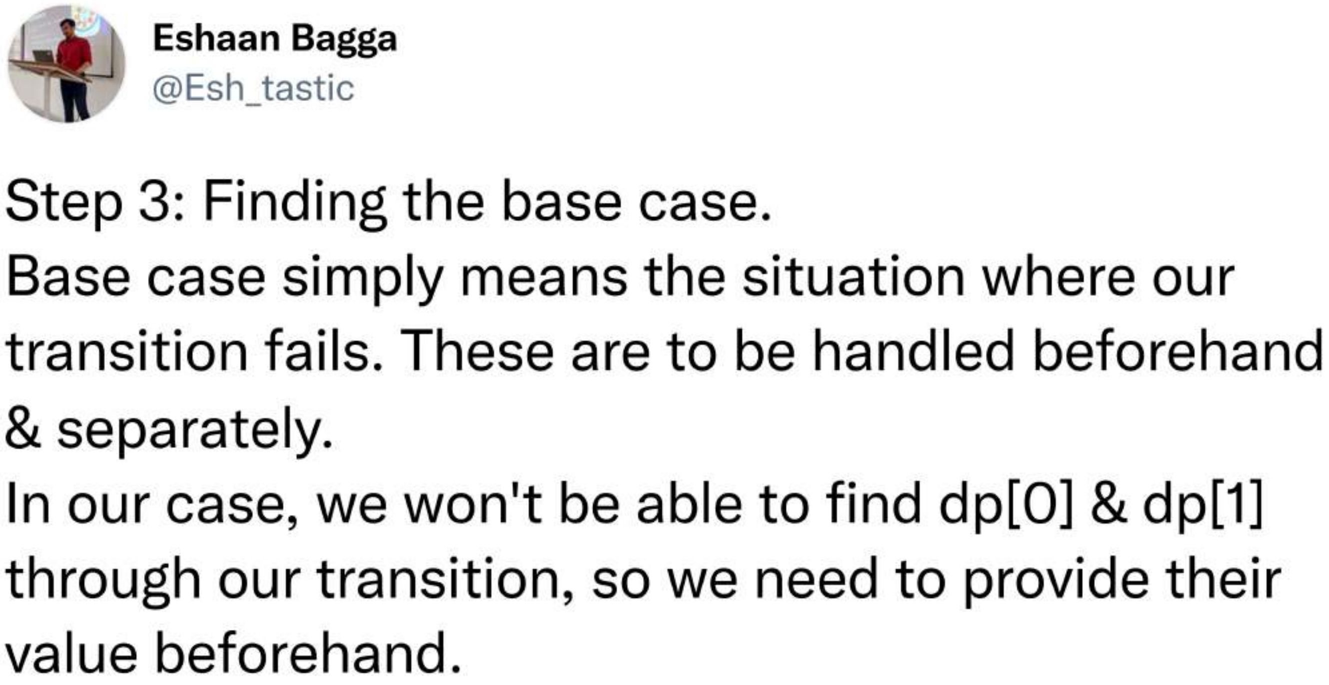
Eshaan Bagga
@Esh_tastic

Step 1: Clearly define the state.
Don't just create a 'dp' array to solve the problem.
Clearly define when you say dp[i] or dp[i][j], what it actually means.
In our case, dp[i] signifies the ith Fibonacci number.



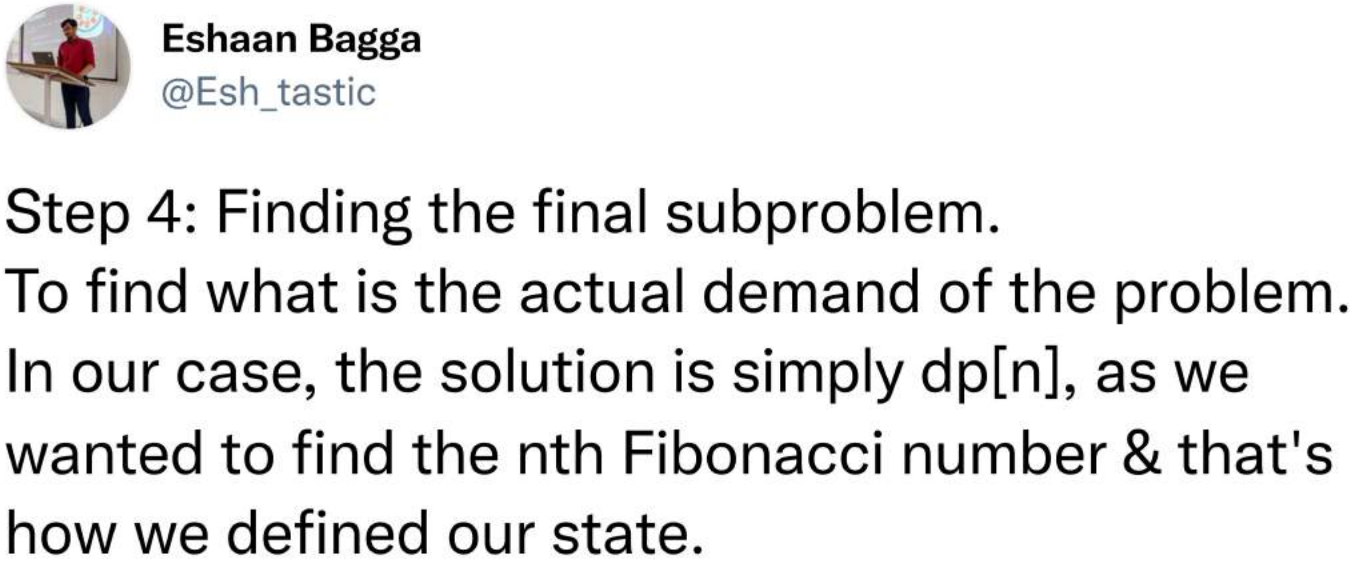
Eshaan Bagga
@Esh_tastic

Step 2: Find the transition.
Transition is simply a relationship between states.
Assume that the states on the right-hand side of the equation have already been calculated.
In our case, the transition would simply be dp[i] = dp[i-1] + dp[i-2].



Eshaan Bagga
@Esh_tastic

Step 3: Finding the base case.
Base case simply means the situation where our transition fails. These are to be handled beforehand & separately.
In our case, we won't be able to find dp[0] & dp[1] through our transition, so we need to provide their value beforehand.



Eshaan Bagga

@Esh_tastic

Step 4: Finding the final subproblem.
To find what is the actual demand of the problem.
In our case, the solution is simply dp[n], as we wanted to find the nth Fibonacci number & that's how we defined our state.

Eshaan Bagga
@Esh_tastic

Credits to @Priyansh_31Dec sir for sharing this knowledge in his IIT GN DP Bootcamp🙌
Link to the Bootcamp: bit.ly/3jew6dA

Dynamic Programming Bootcamp | Day 1/6 | Taught at IIT Gandhinagar | Competitive Programming

🔗 youtube.com