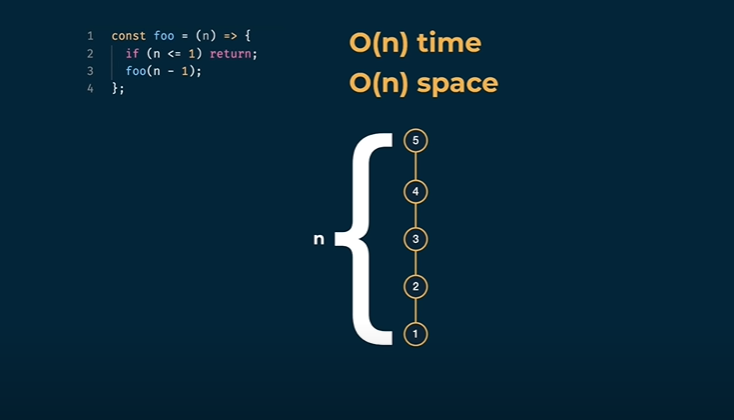
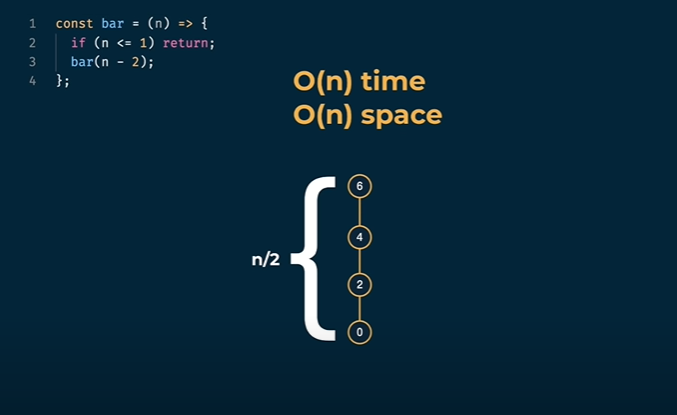
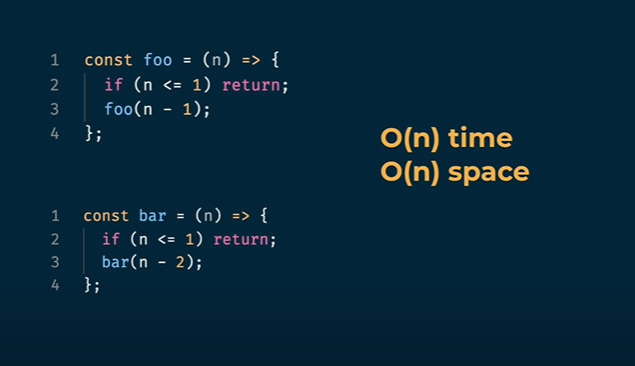
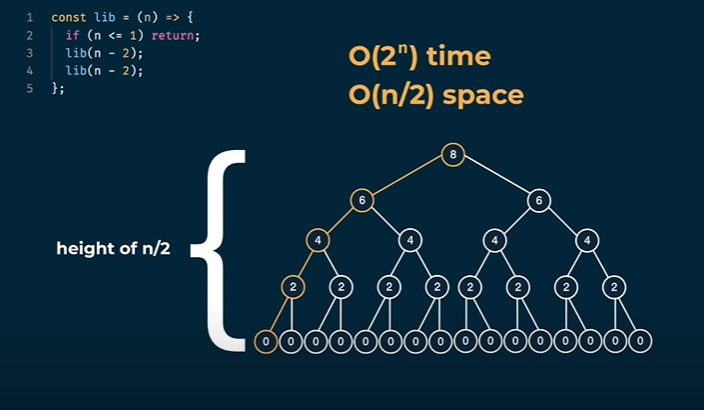
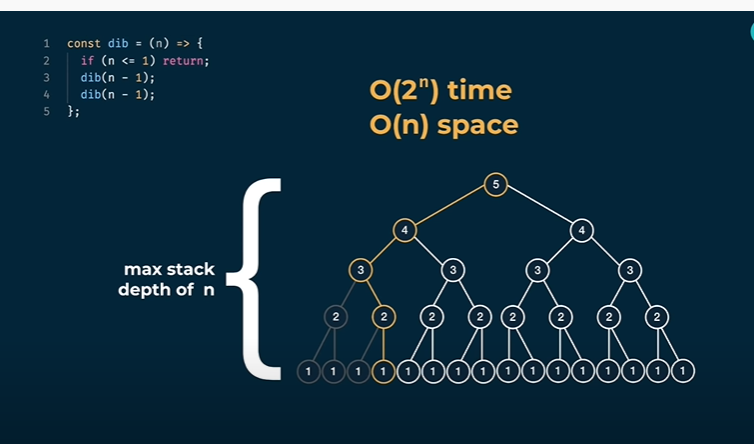
**Important Links 🡪** [**YT Course Link**](https://www.youtube.com/watch?v=oBt53YbR9Kk&t=9539s)[**My Github Link**](https://github.com/Mohit991/DP_Programs/tree/main)[**Course Summary on Github**](https://github.com/irfan7junior/algorithms-in-java)

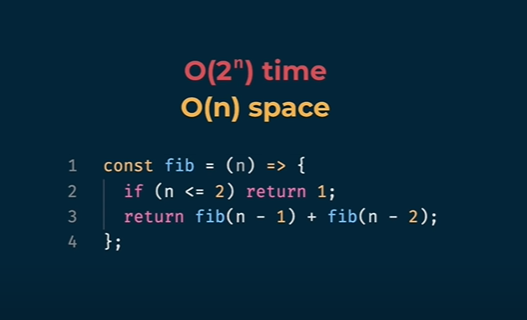
**Time Complexity:**



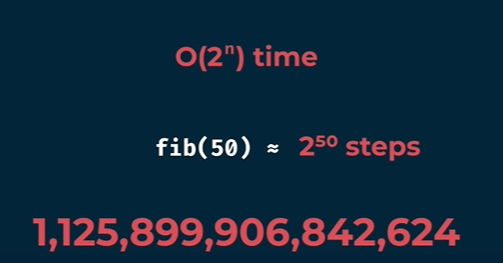






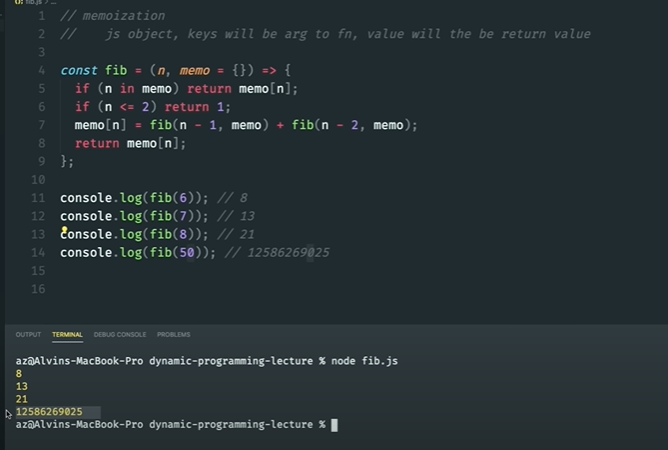


Because right now we are not using DP.

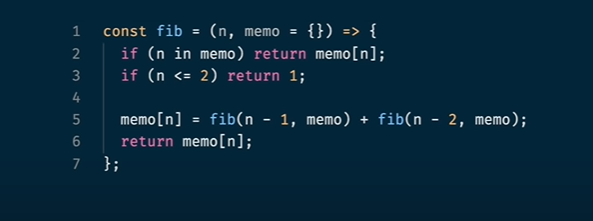


Very large time.

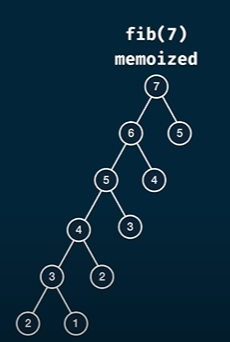
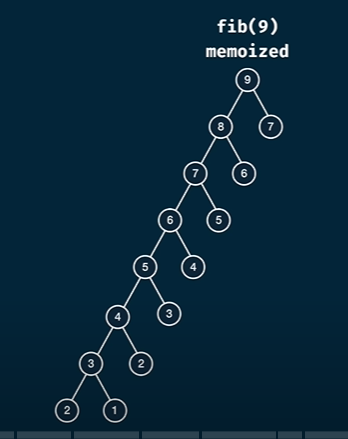
JS Code:



We are using memorization and storing the result of previous ones and using them later.

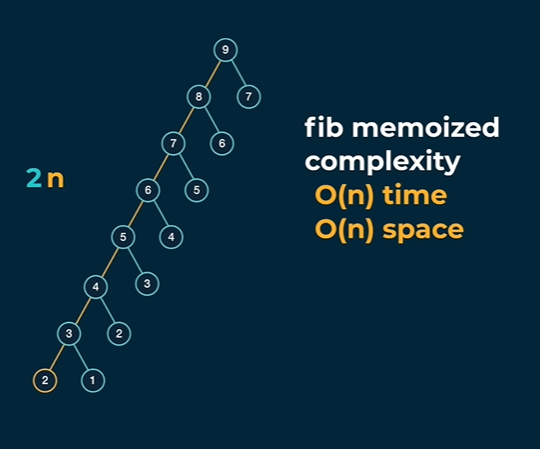


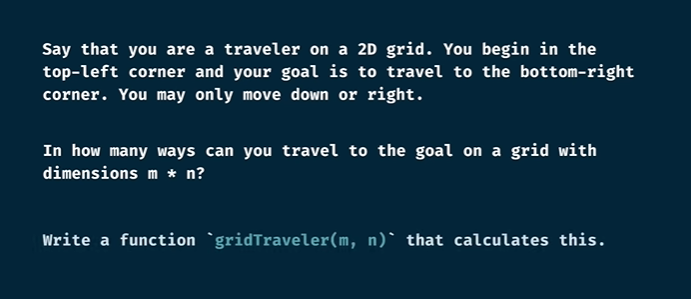
With DP:

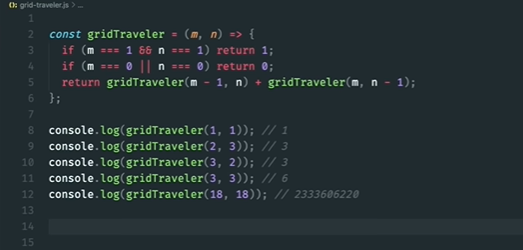
And so on….

TC for memorization:

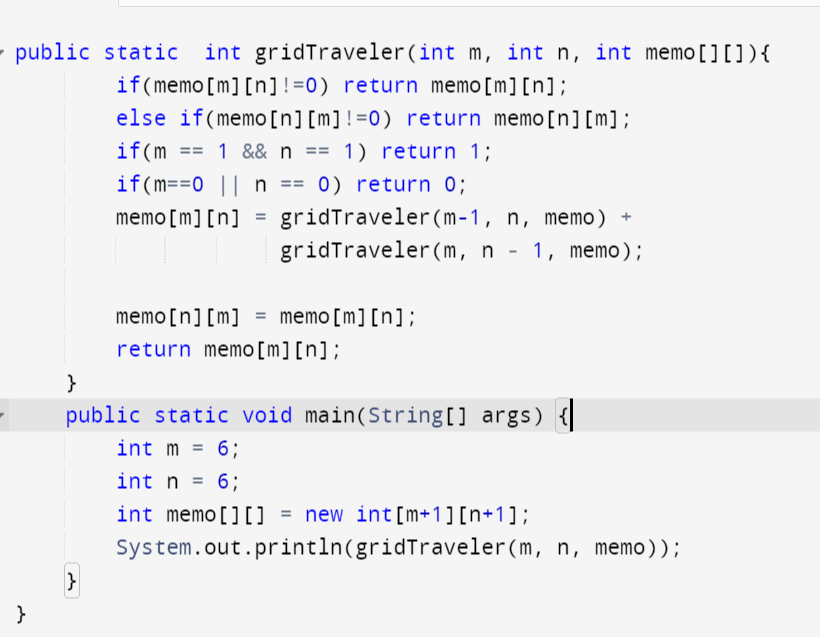




In JS(Without DP):



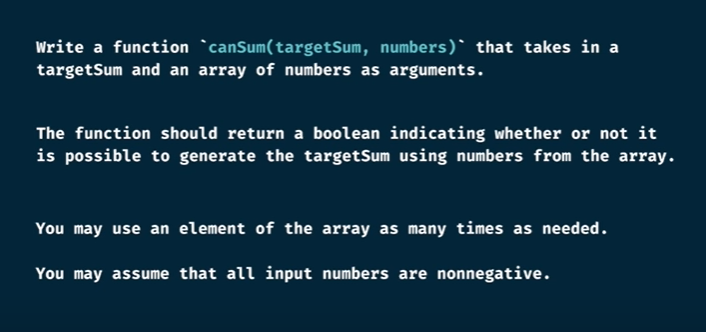
With DP(Java):



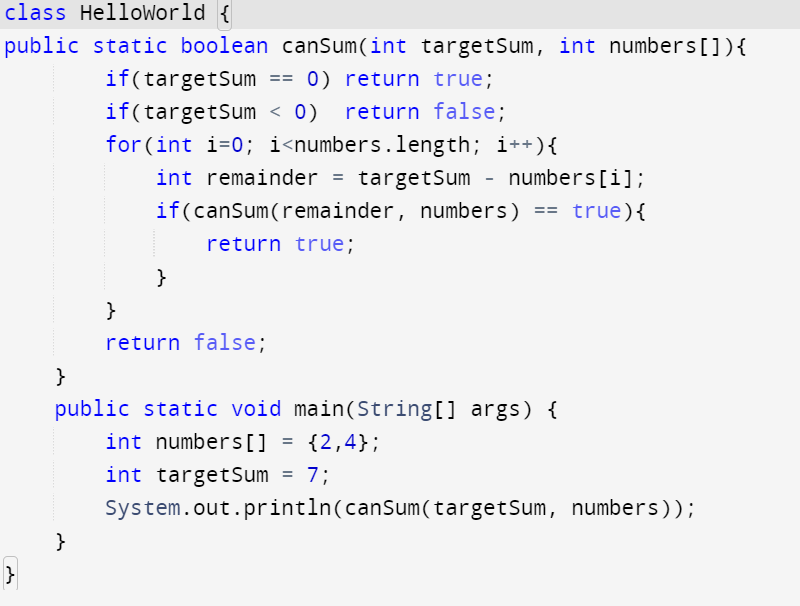
How to do memorization:



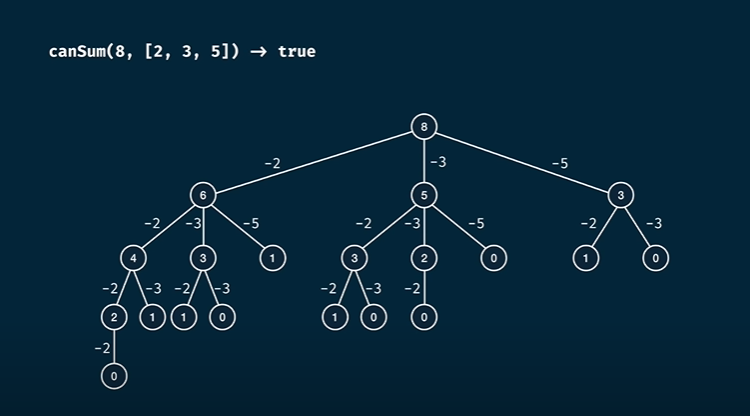
Problem:

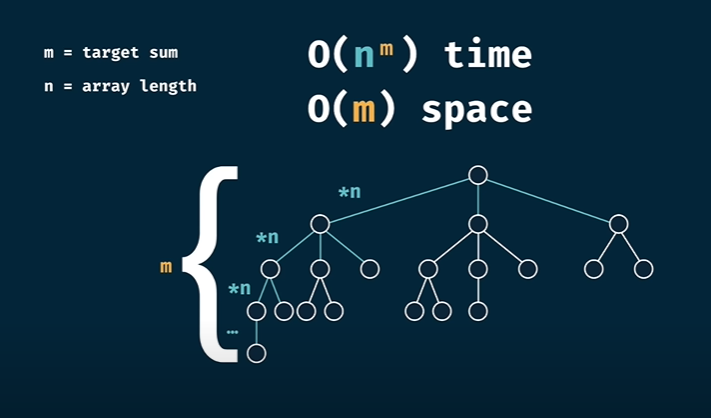


Without DP:

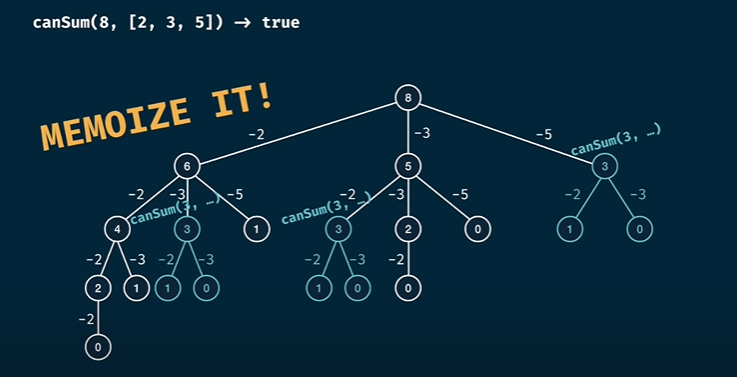


TC:

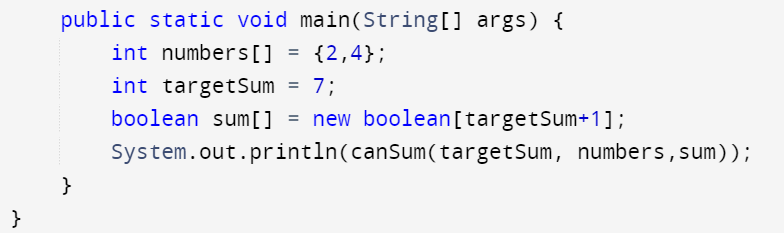
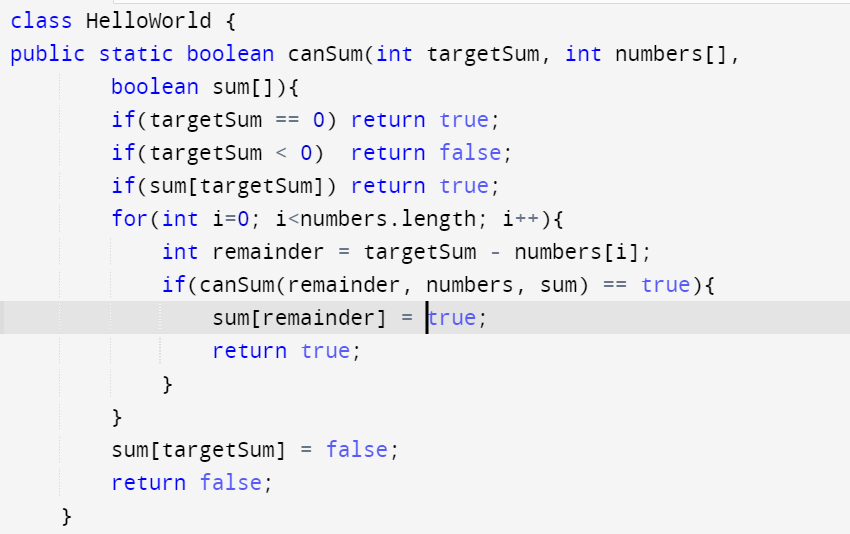




To use DP here. Look at the overlapping sub problems.

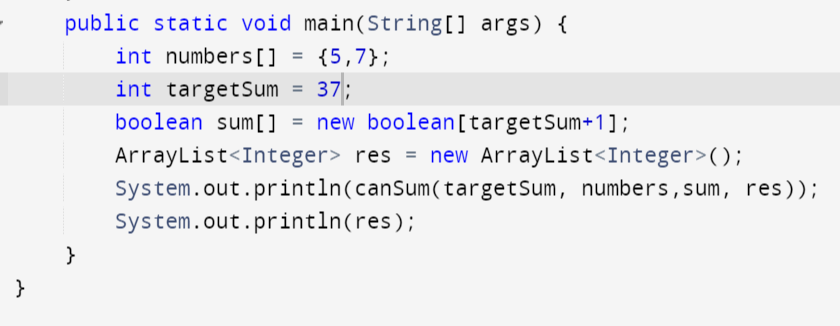
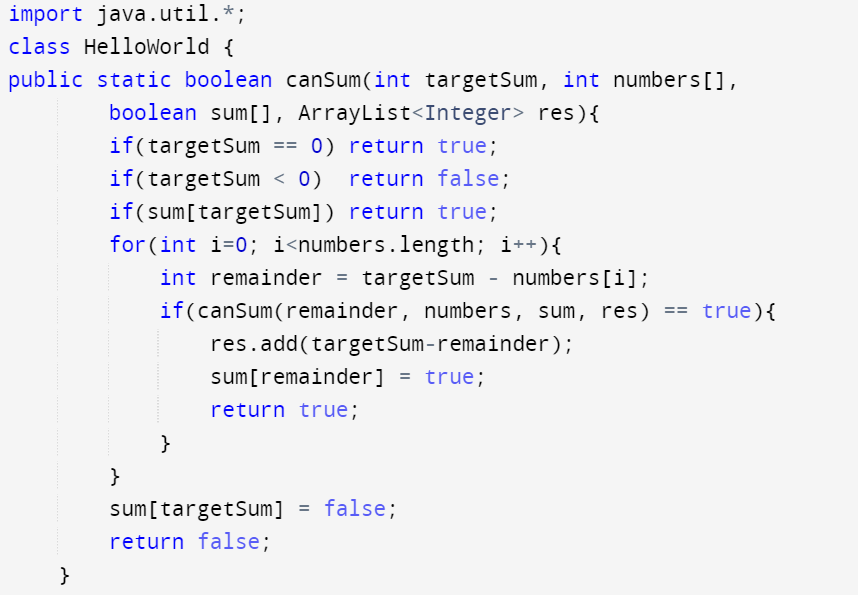


With DP:

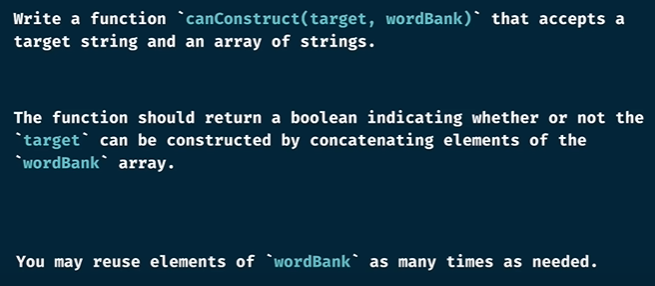


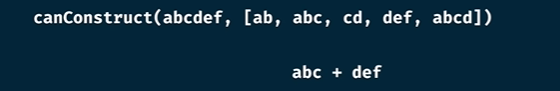


If we also want to know which array returns the result:

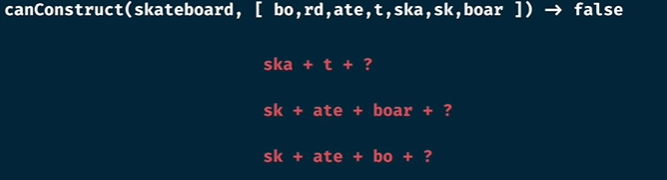


Next Problem:





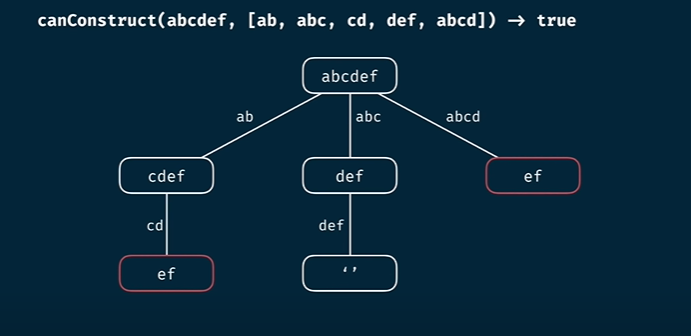




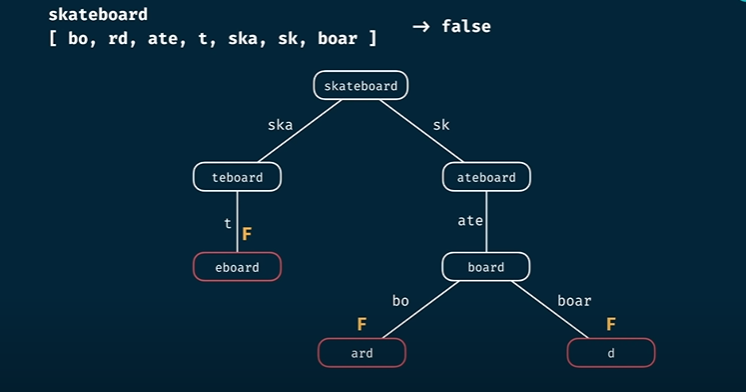
Base Cases:



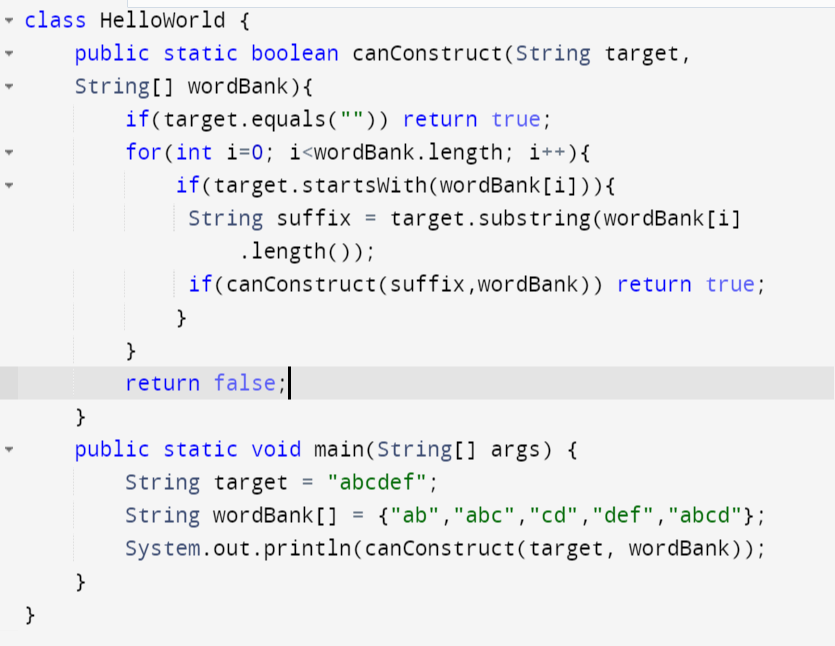
Logic:



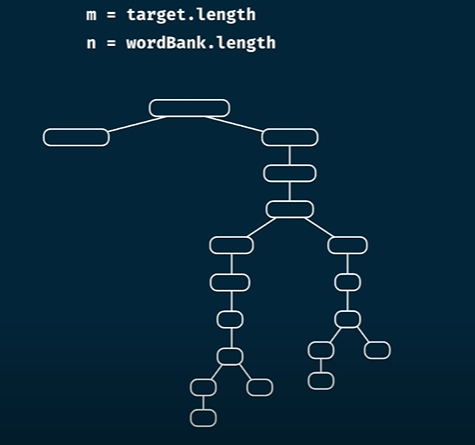
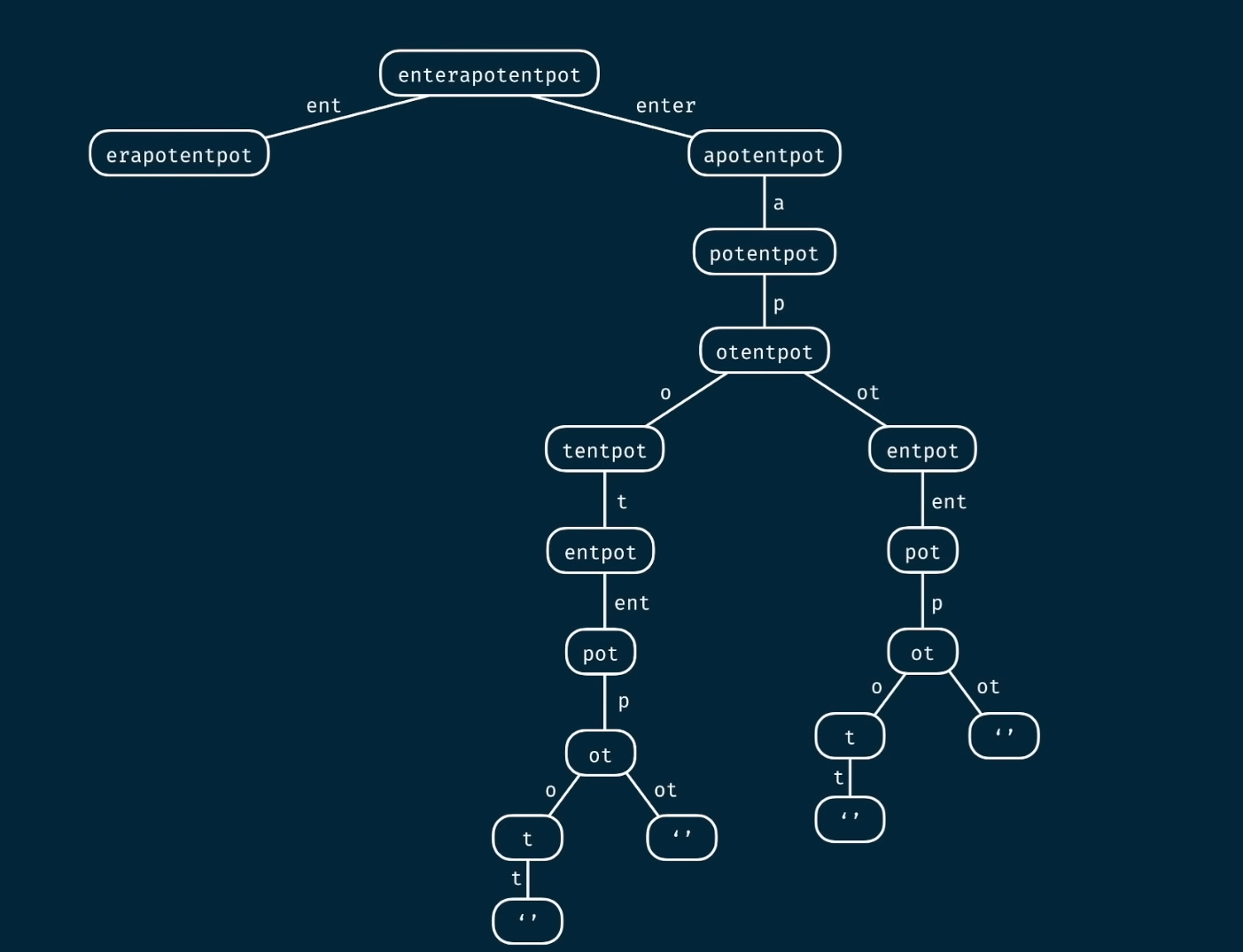
Only take those elements from the wordbank which are prefix to the targetString.



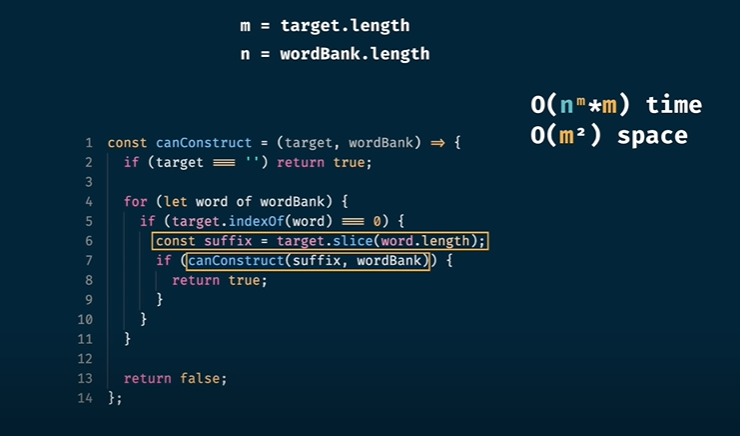
Java Code without DP:



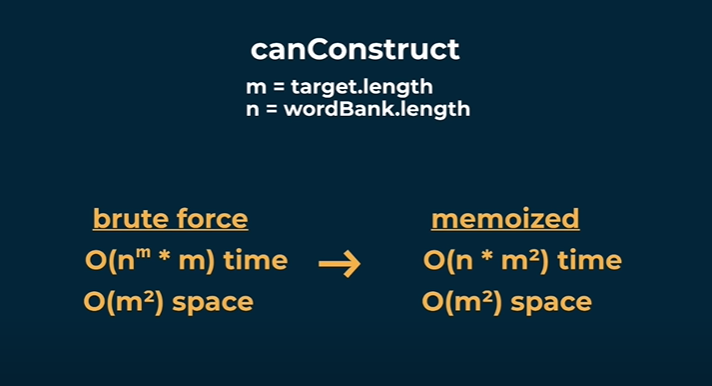
For TC:



TC and SC:



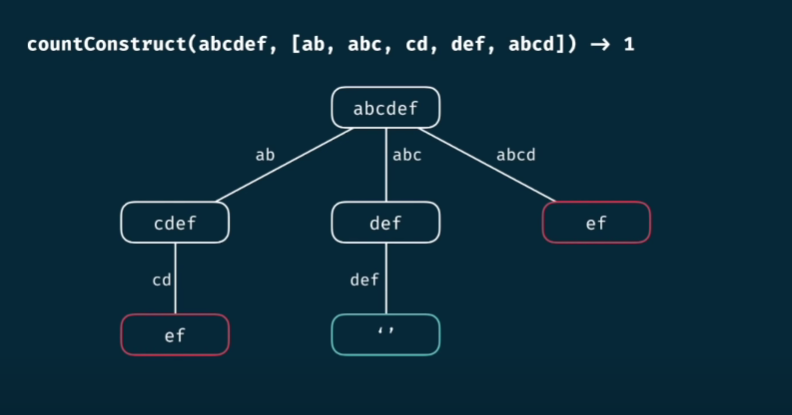
TC with DP:

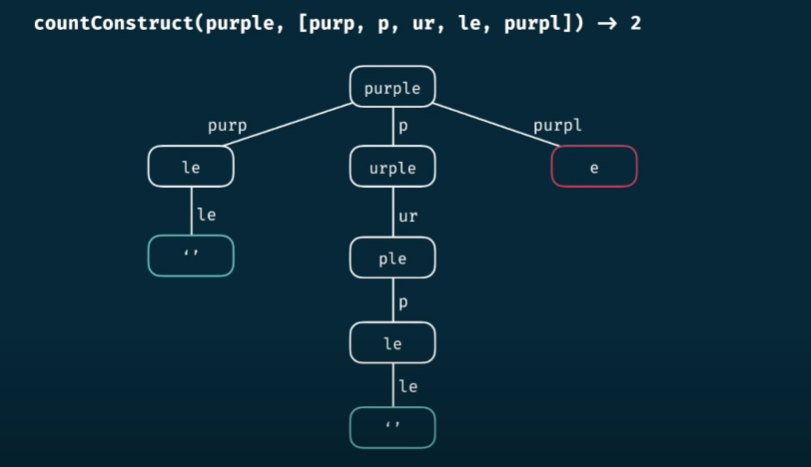


Next Problem:

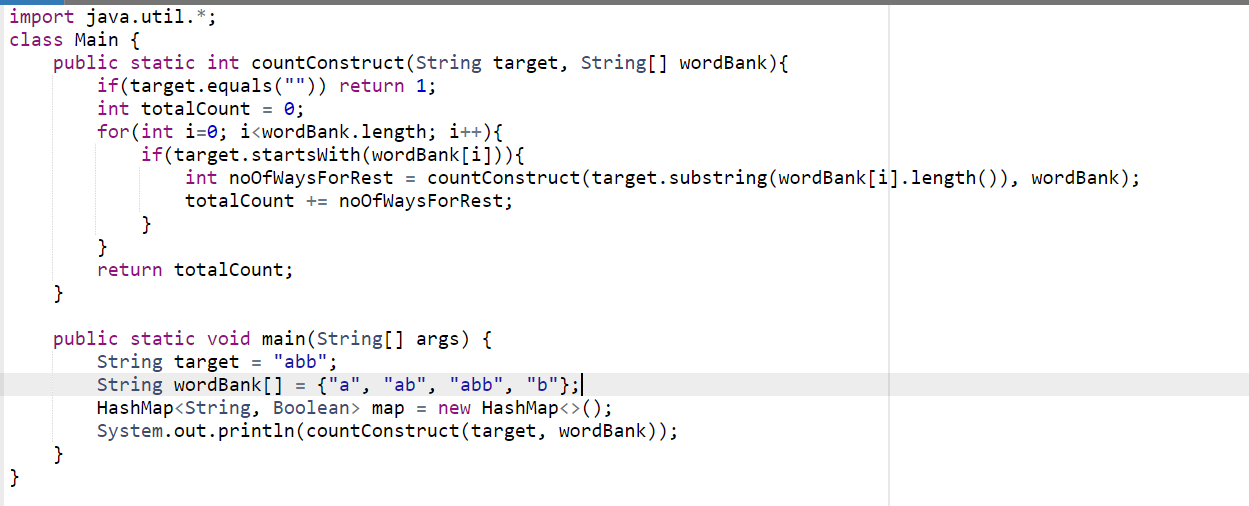


It is same problem but here we need to basically count the no of ways.

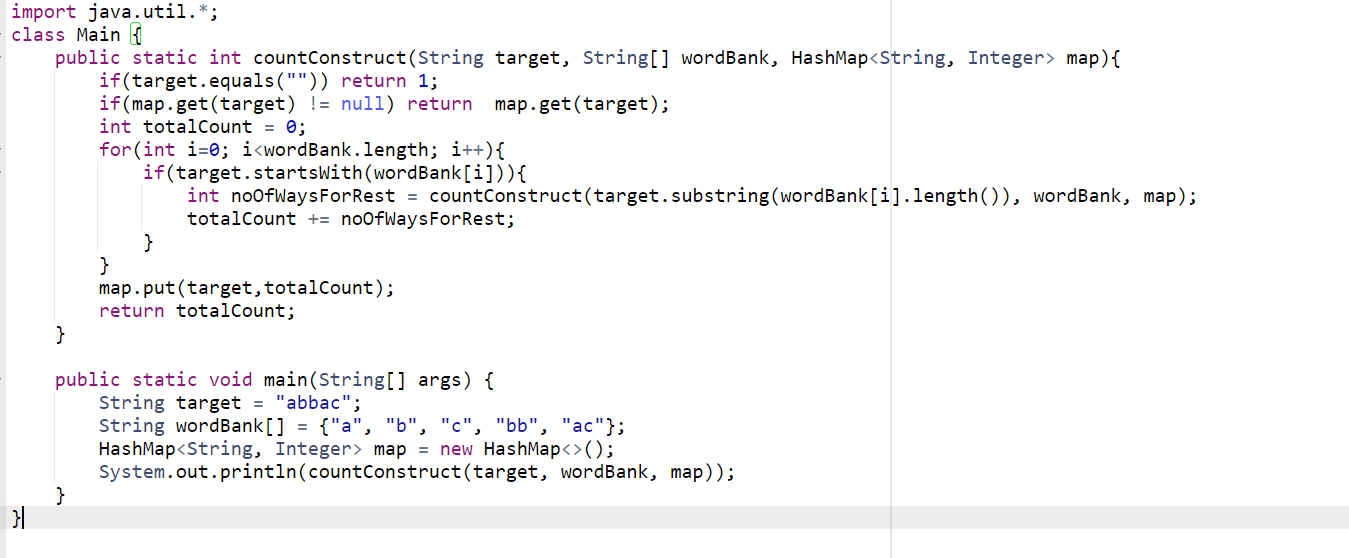
Red will return zero and green will return 1.



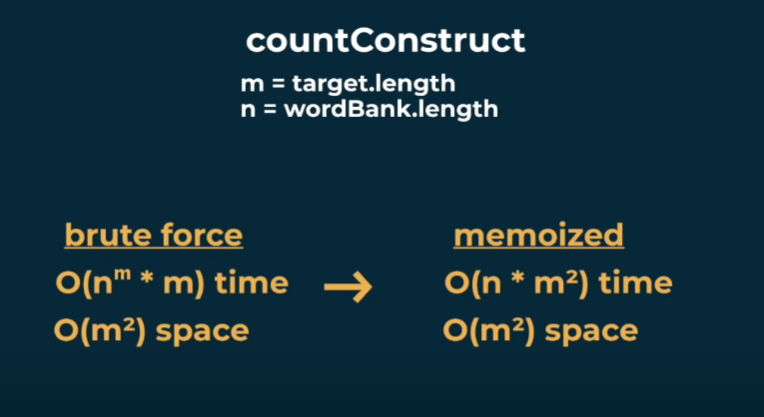
Java Code without DP:



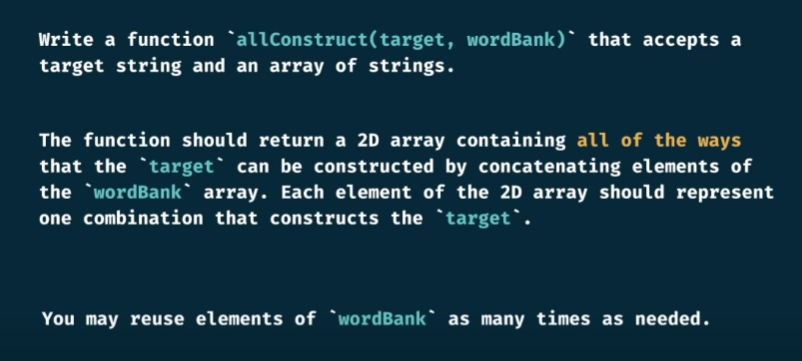
With DP:

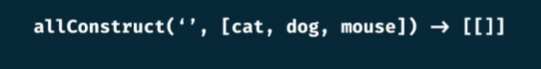
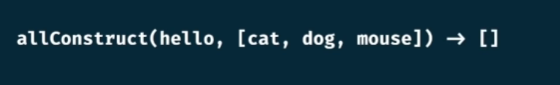
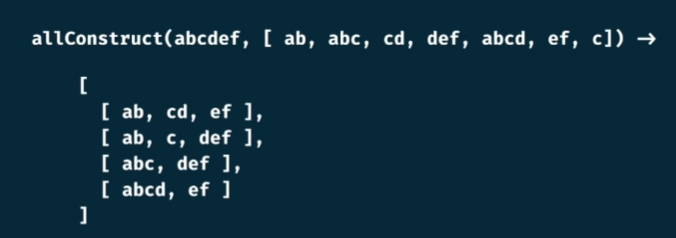
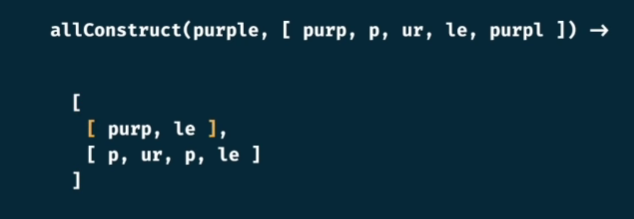


TC:



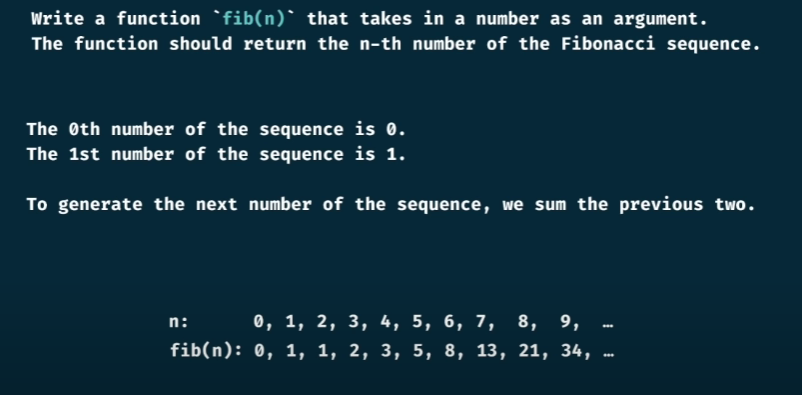
New Problem 🡪



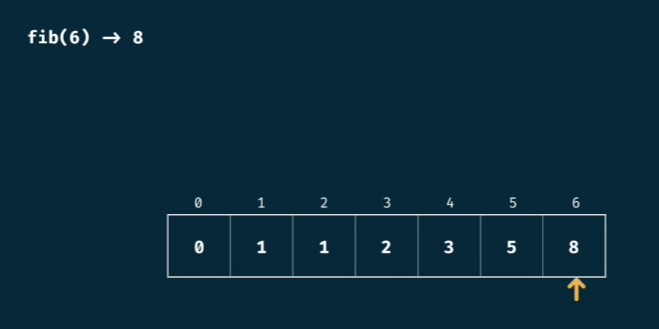


We will solve this problem later.

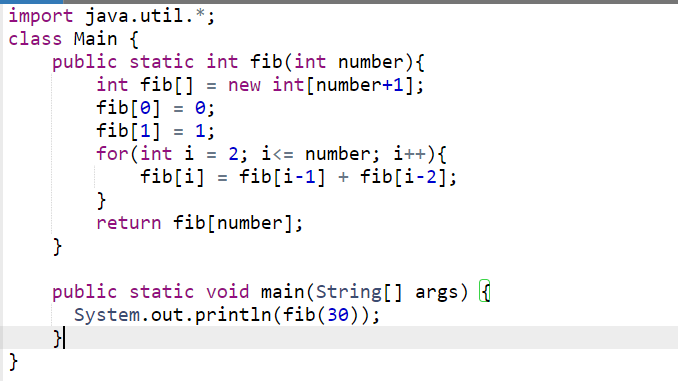
Tabulation 🡪



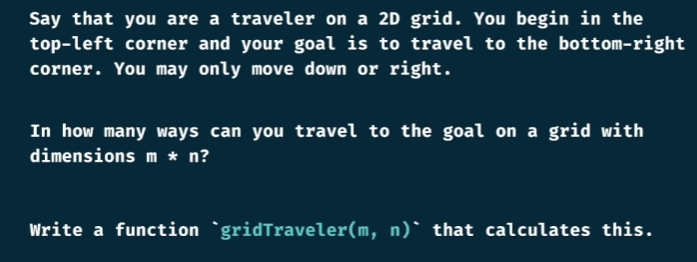
We create an array or a table and store the result in that and then use those result and move on.

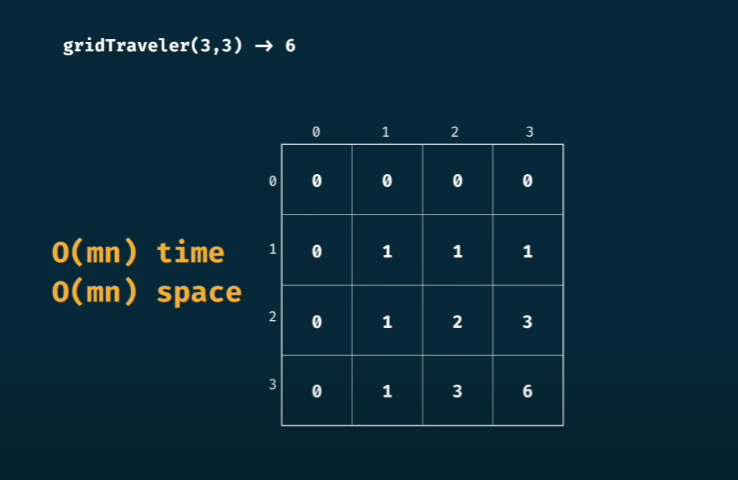


Tabulation Answer 🡪

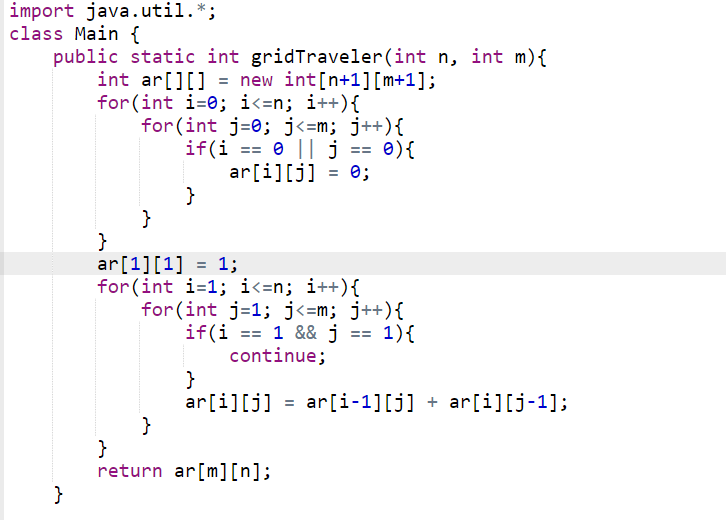


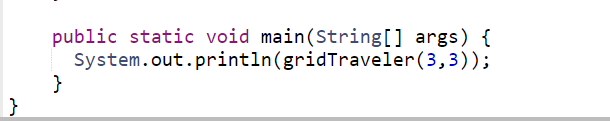
GridTraveler Problem using Tabulation 🡪



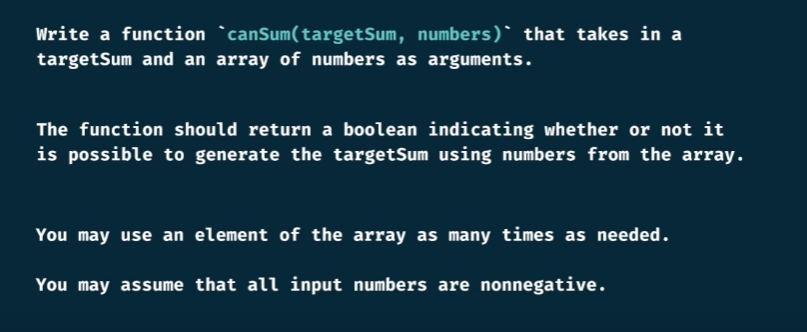


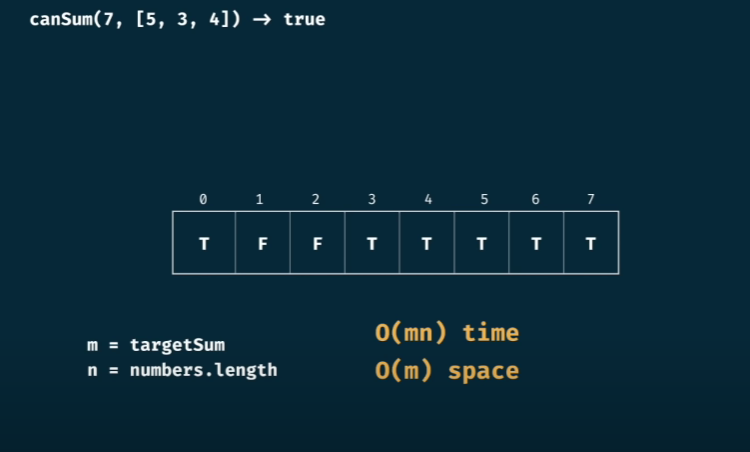
Solution 🡪

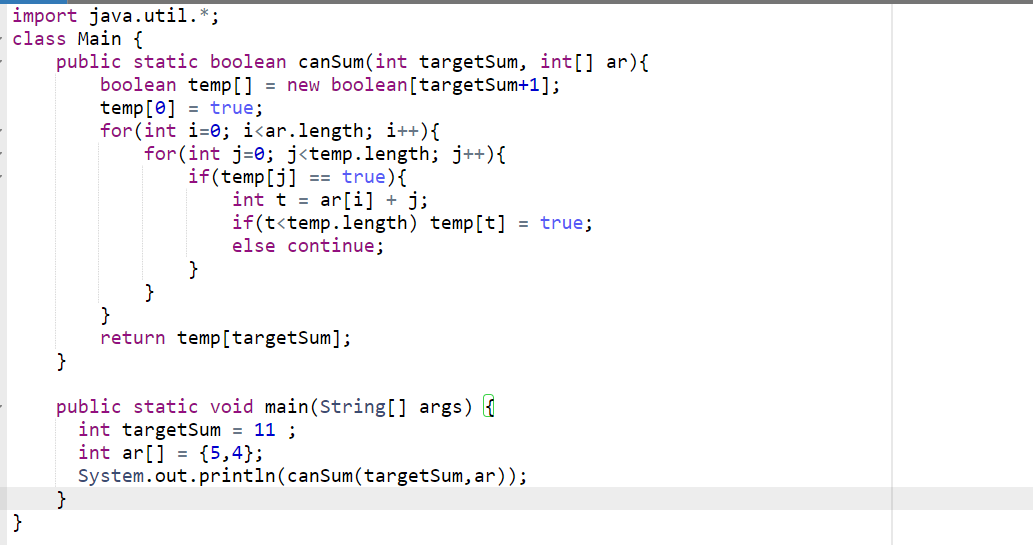




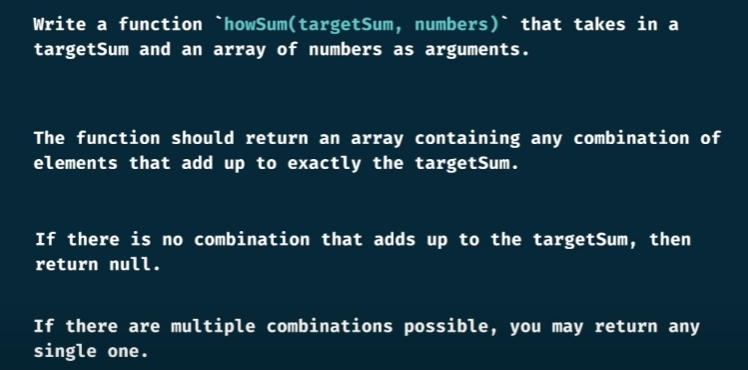


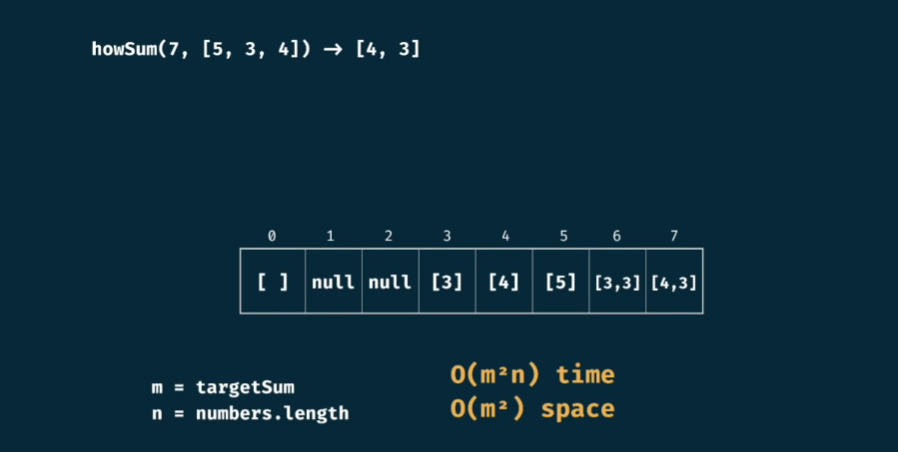
Problem 🡪

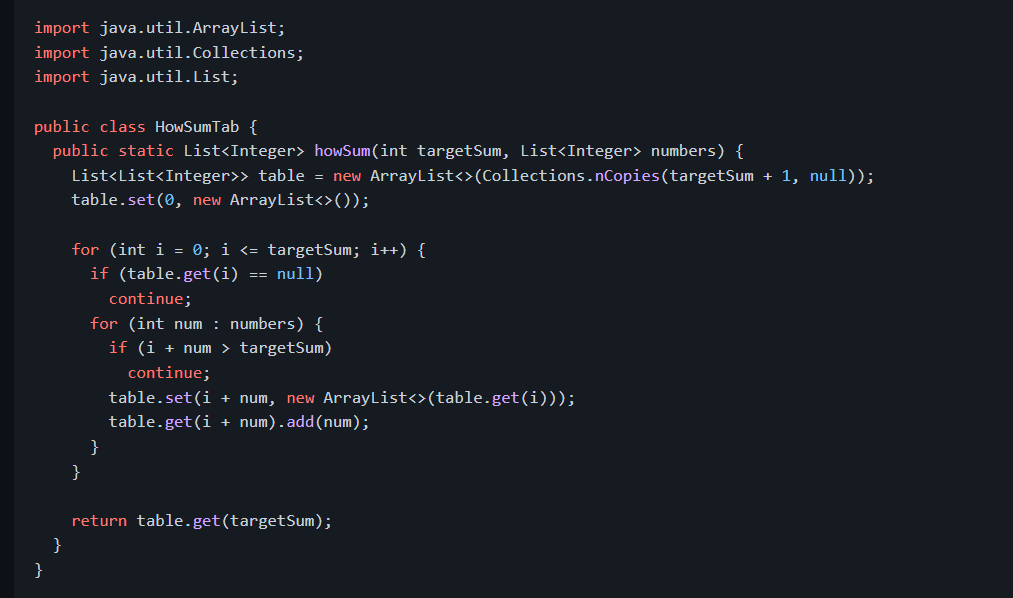




Problem 🡪



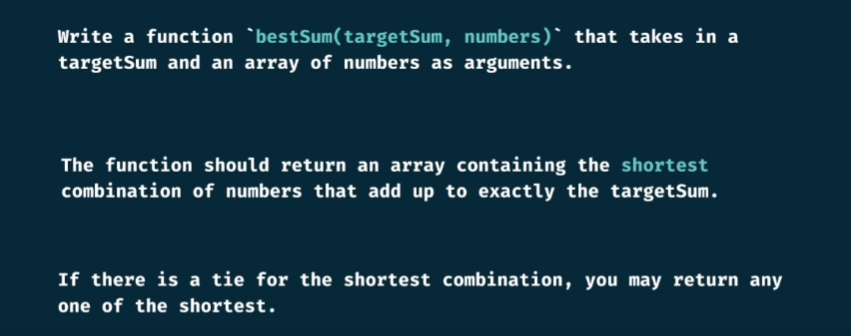


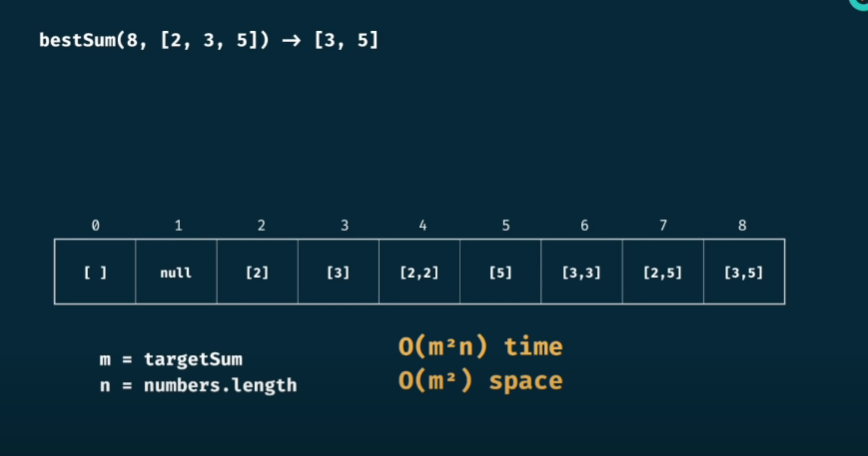


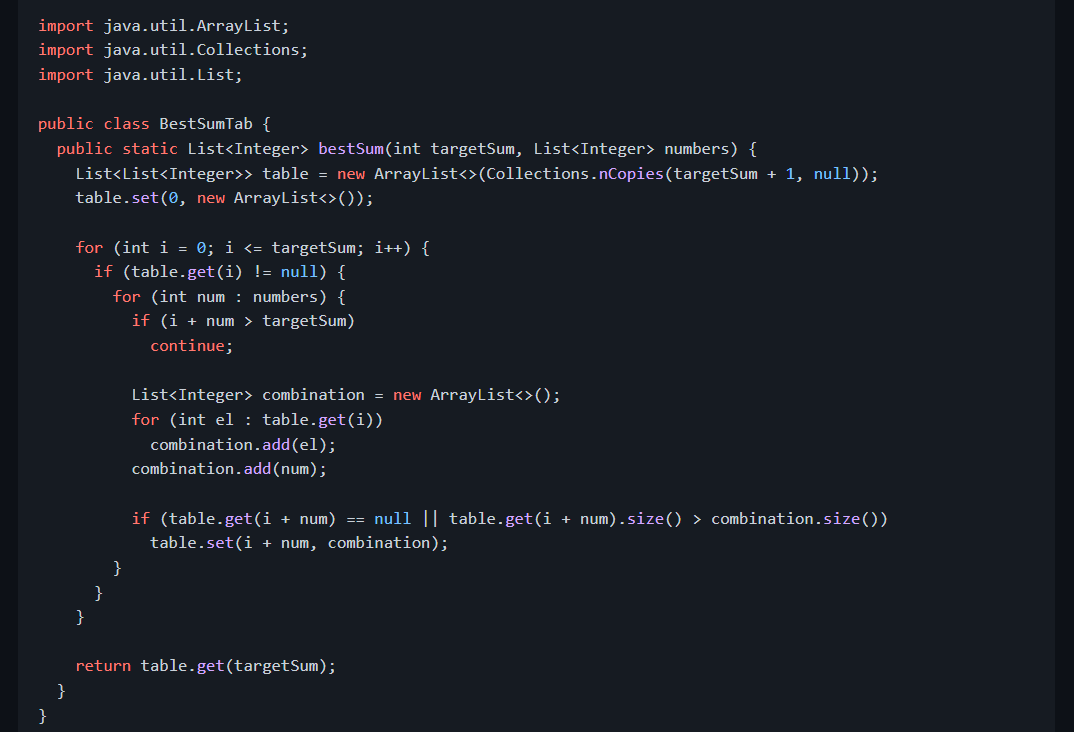
TC = O(m^2 \* n)

M = no of elements

N = length of array

Problem 🡪



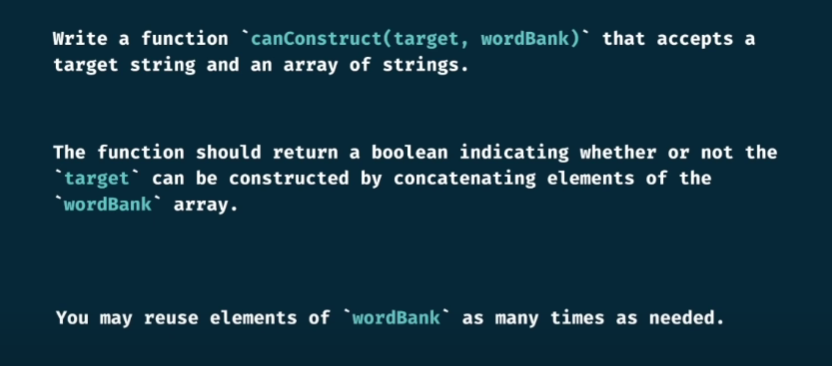


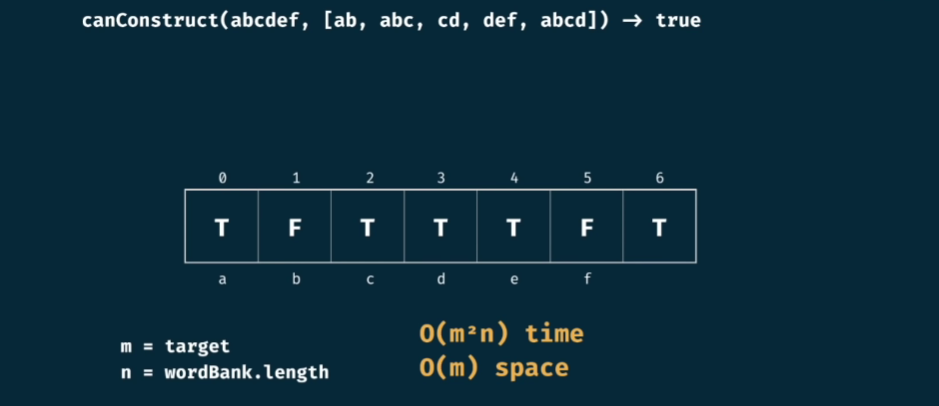
TC = O(m^2 \* n)

M = no of elements

N = length of array

Problem 🡪





FINAL 🡪

**Dynamic Programming**

* notice any overlapping subproblems
* decide what is the trivially smallest input
* think recursively to use Memoization
* think iteratively to use Tabulation
* draw a strategy first!!!