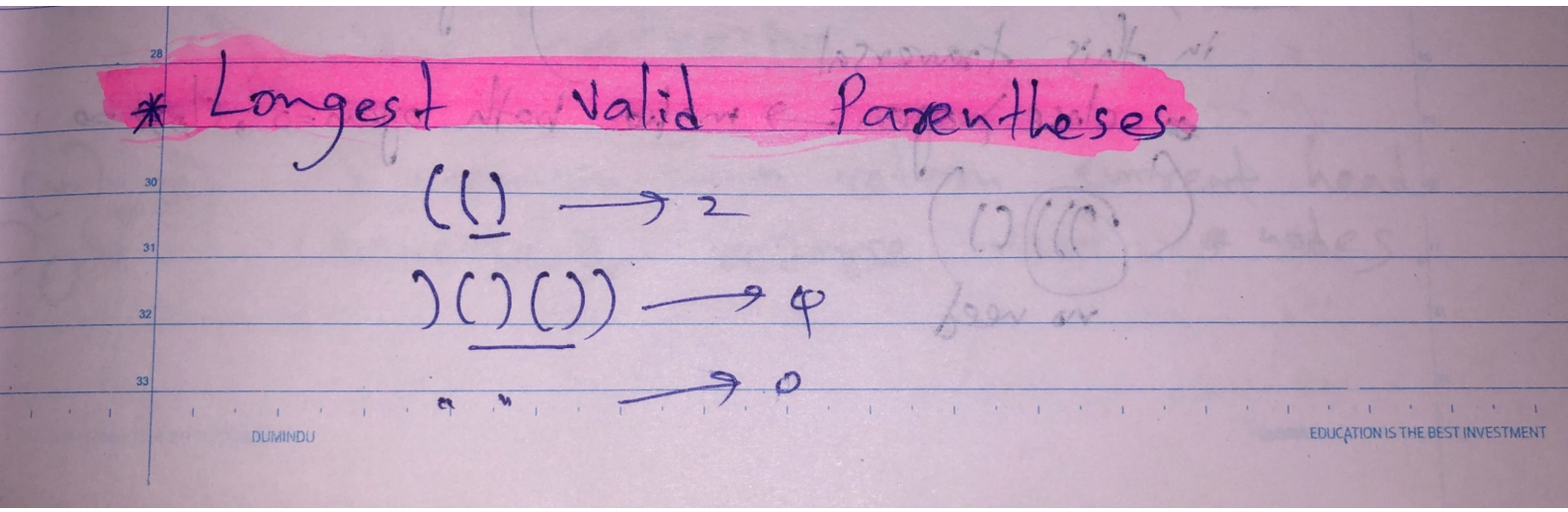

$O(n^3)$ $O(n)$ Bruteforce → from each place find all possible substrings & then calculate validity & take max.

$O(n)$ $O(n)$ dp → ()() → $dp[i] = dp[i-2] + 2$

((()())) → $dp[i] = dp[i-1-dp(i-1)] + dp[i-1]$

stack

((()())) ()

$O(n)$ $O(n)$

(→ push to stack

) → pop top → take difference from remaining to current

if empty → push index

else → take max from index-peek

((()())) () traverse on both sides counting open & close ()

→ in this traversal

* close > open → make both open=0, close=0

( ))) ()

no need

← do this side of traversal

)((( )))(((

if open brackets > closed,
then make both counts to 0.

take max of both traversals of both sides.

when open = close, take 2 * open as matching pairs count

O(n)
O(1)

((

→ we don't find 2 as max count.
so we do ← traversal.