

Input: string  $s$

actions on  $s$ : remove any pair of consecutive characters

Output: Lexicographically smallest string

Main idea: Solve with 2 recursive subroutines

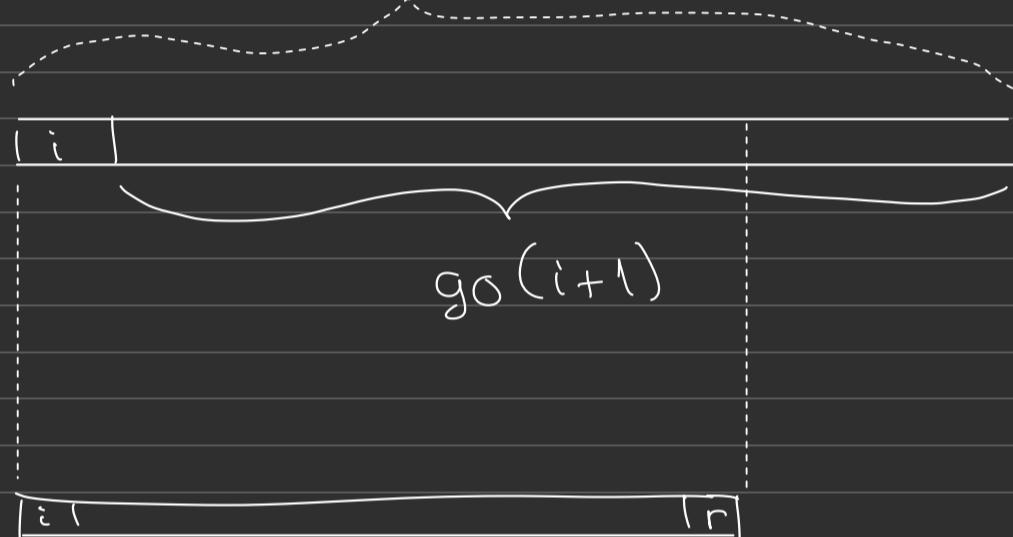
def  $g(i) \rightarrow \text{string}$ : what is the "best" string  $s$  ?

def  $\text{canRemove}(l, r)$  can this substr be removed?

Example:

$s[i:] + g(i+1)$

$g(i):$



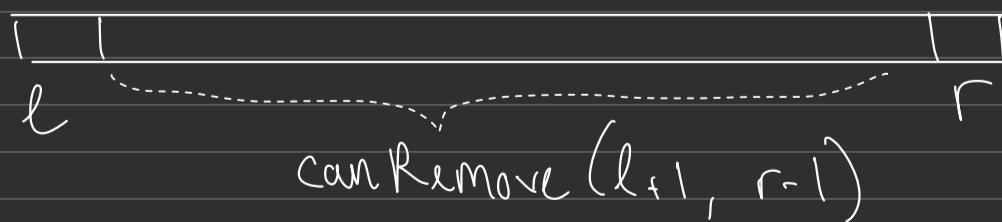
for ( $r: i \rightarrow n$ ):

$\text{canRemove}(i, r)$

return  $\min \begin{cases} s[i] + g(i+1) & \rightarrow \text{original substr} \\ g(r+1) & \forall r \in [i, n] \end{cases}$

$\text{canRemove}(l, r):$

Case 1: remove the middle



Case 2: remove either  $l$  portion or  $r$  portion

