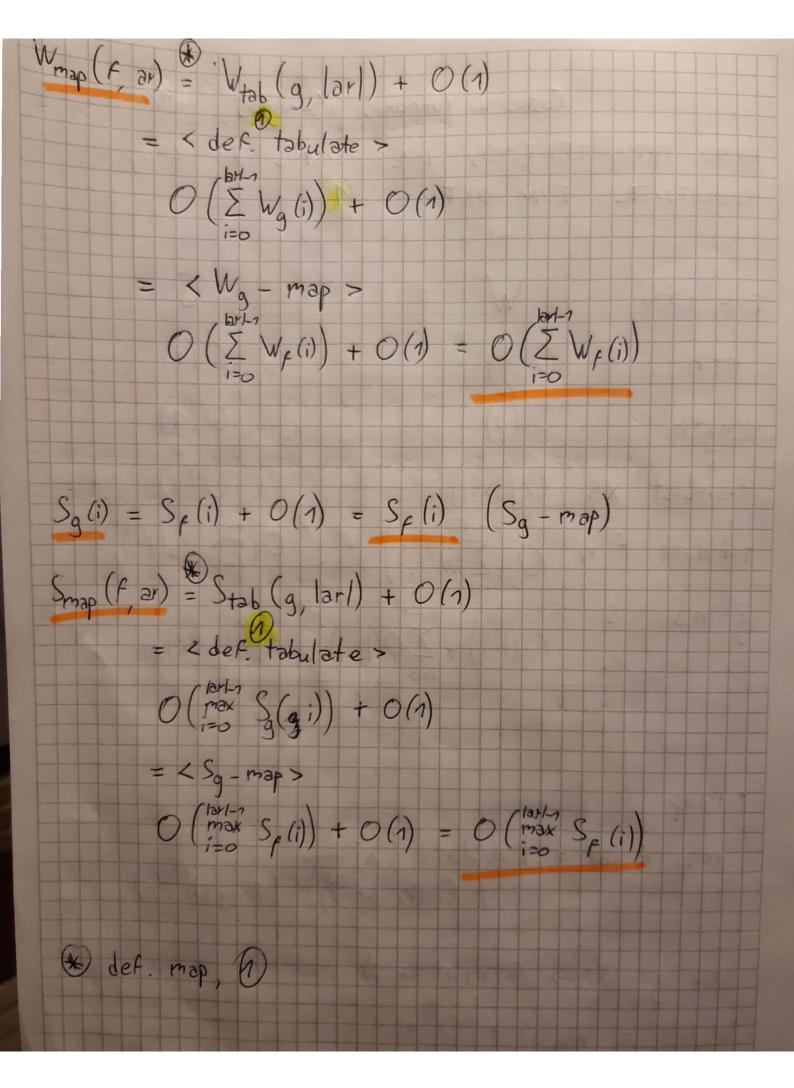
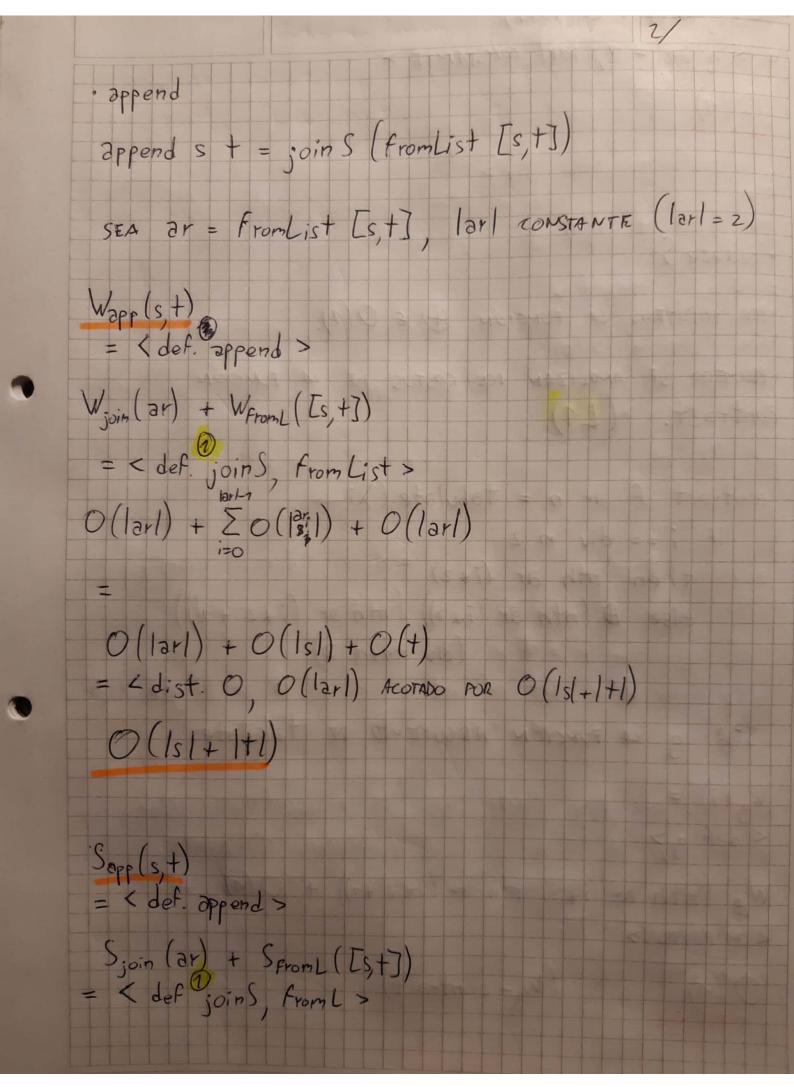
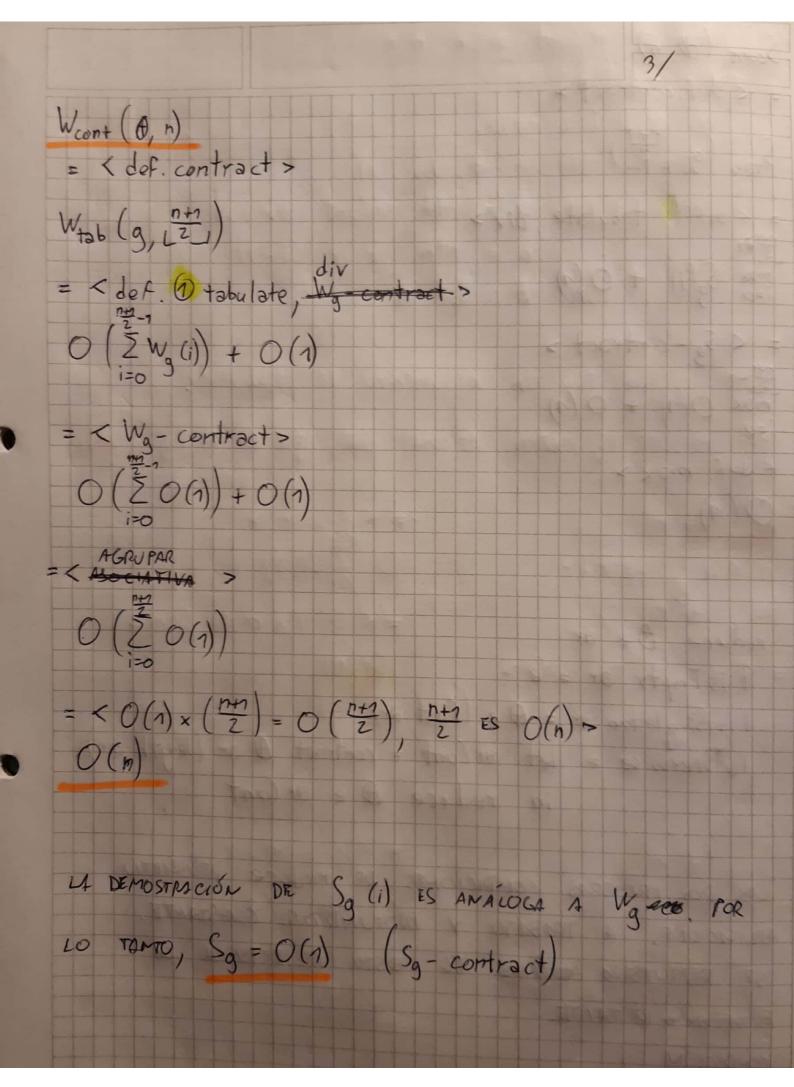
COMO DATO,	Costos de funciones plavreglos 1/ TENEMOS LOS COSTOS DE LAS FUNCIONES: 1)
FUNCIÓN	0(1) O(1)
length FromList XS	O(1) $O(1)$ $O(1)$ $O(1)$
tabulate f n	$O\left(\sum_{i=0}^{n-2}W(F_i)\right) O\left(\sum_{i=0}^{n-2}S(F_i)\right)$
	$O(s) + \sum_{i=0}^{ s -1} O(s_i) O(s s)$
map far	= tabulate (\lambda i -> F (nth ar i)) (length ar)
W ₀ (i) = W ₀	> F (n+h ar i) (i) + O(n)
= < Wp	(i) (i) ACOTADO POR Wp(i) >

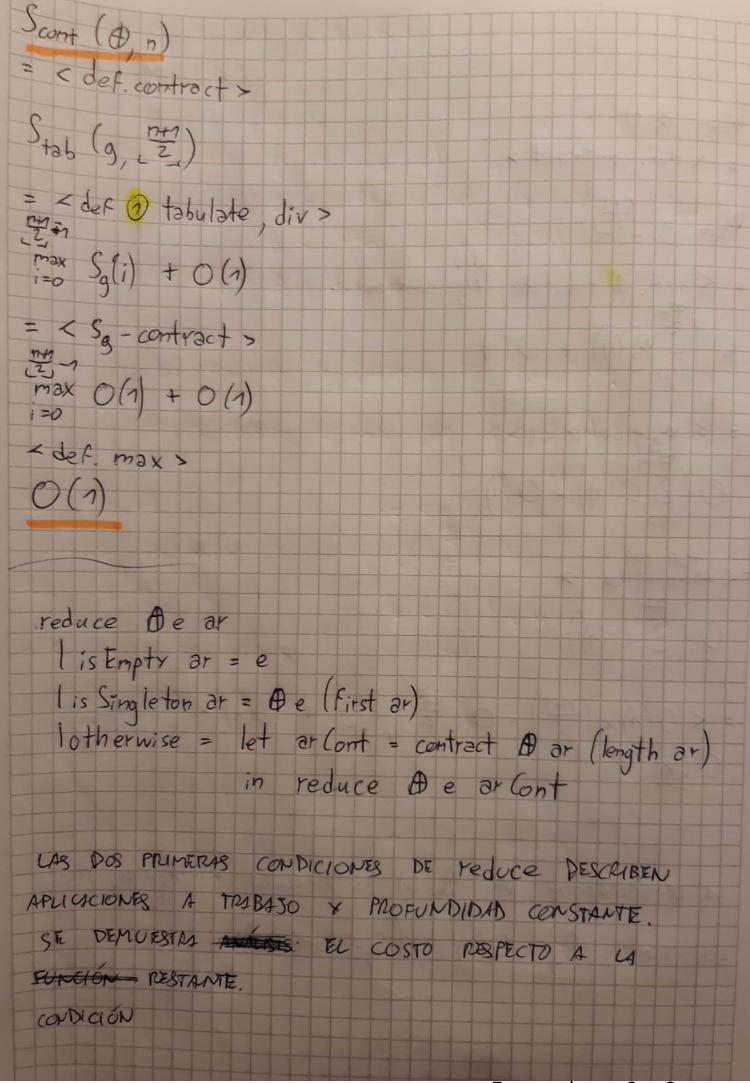




O(1g (arl) + O(1arl)
O(lg larl) + O(larl) = < lg larl = 1, larl constanth >
0(1)
· reduce S
POR MIPÓTESIS LA FUNCIÓN (CO (1).
PRIMERO, SE AMALIZAN LOS COSTOS DE LA FLACIÓN
contract. (L1)
contract f ar n = tabulate ():->
if i == div n 2
then pth ar (i * 2)
else \oplus (nth ar (i*z)) (nth ar (i*z +1)) (div (n+1) z)
SEA 9 4 FUNCIÓN ARGUMENTO DE tabulate.
W _g (i) = n+h
< def 9 ? (1 1) () () () () () () () ()
Wa (nth ar (i*z), nth ar (i*z+1)) + O(1) z MIPÓTESIS >
O(1) (Wg-contract)
Escaneado con CamScanne

Escaneado con CamScanner





Escaneado con CamScanner

Wred (B, ar) = < def. reduce > Woon+(+, larl) + Wred (+, 2) = < def contract> O(n) + Wred (1) (2) = < TEOPEM MARSTRO > O(n) Sred (ar) = < def. reduce > Scont (1) ar) + Sred (1) = < def. contract> < L1> O(n) + Sred (2) O(la larl) · scan S POR HIPÓTESIS, LA FUNCIÓN DEO(1). PRIMERO, SE ANALIZAN LOS COSTOS DE LA FUNCIÓN expand (LZ)

expand of ar Cont ar Or n tabulate (x1-> if mod i 2 == 0 then nth ar Cont (div i 2) else let a = nth ar Cont (div i 2) in $\beta = nth \text{ ar Or } (i-1)$ SEA Q LA FUNCIÓN ARGUMENTO DE tabulate. POR MIPOTESIS, PODEMOS AFIRMAR QUE 9 8 0(1) (q-scans) Wexp (.+, n) = < q -scans> 0 (20(1) = < def. expand > Wal (n) = < O(1) x n Es O(n) s O(n) = < def. 1 tabulate > Sexp (D, n) = < g - scan 5 > = < def expands O (max O(1)) Sexp (g, n) = < def. (1) tabulate > = < def. max> O (max Sq (i))

scans de ar * lis Empty ar = (ar, e) lis Singleton ar = let el = first ar in (singleton e, De el) otherwise = let nilem = length ar er (ont = contract f ar nElem (r,t) = scan & e ar Cont in (expand @ r ar nElem, t) IGUM QUE EN reduceS, SE DEMUESTRA EL COSTO PARS LA ÚLTIMA CONDICIÓN DE LA FUNCIÓN SCANS. Wscan (1 n) = < def, scans> When (ar) + Woont (1) n) + Wscan (1) (2) + Wexp (1) = < def @ length, L1, L2 > O(1) + O(n) + Wscan (.0, 2) = < O(1) ACOTADO POR O(n) > O(n) + Wscan (0, 2) Z < TEOREMA MESTRO > 0 (n)

