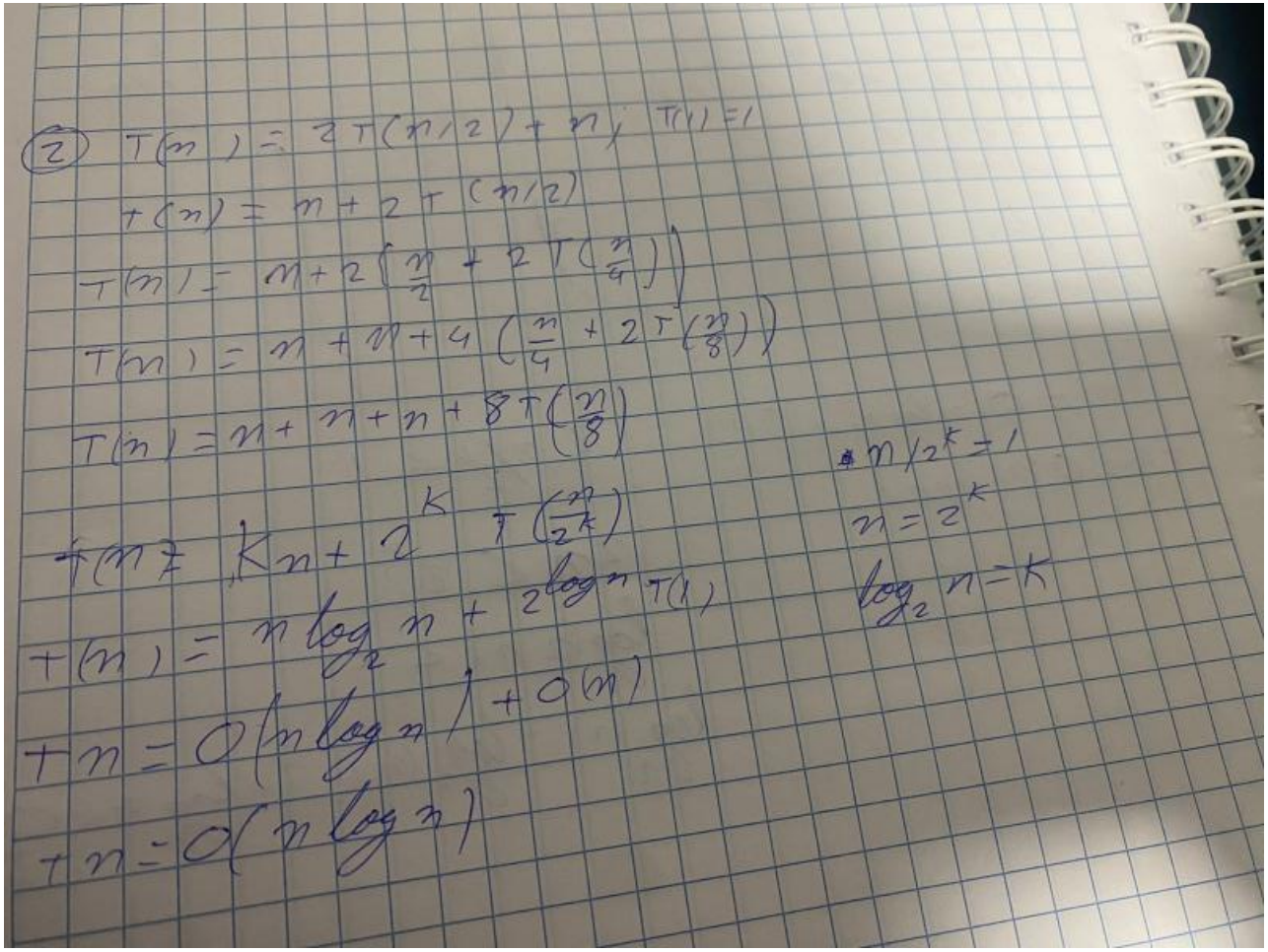


(2) $T(n) = 2T(n/2) + n$, $T(1) = 1$

$T(n) = n + 2T(n/2)$

$T(n) = n + 2\left[\frac{n}{2} + 2T\left(\frac{n}{4}\right)\right]$

$T(n) = n + n + 4\left(\frac{n}{4} + 2T\left(\frac{n}{8}\right)\right)$

$T(n) = n + n + n + 8T\left(\frac{n}{8}\right)$

$T(n) = kn + 2^k T\left(\frac{n}{2^k}\right)$

$T(n) = n\log_2 n + 2^{\log n} T(1)$

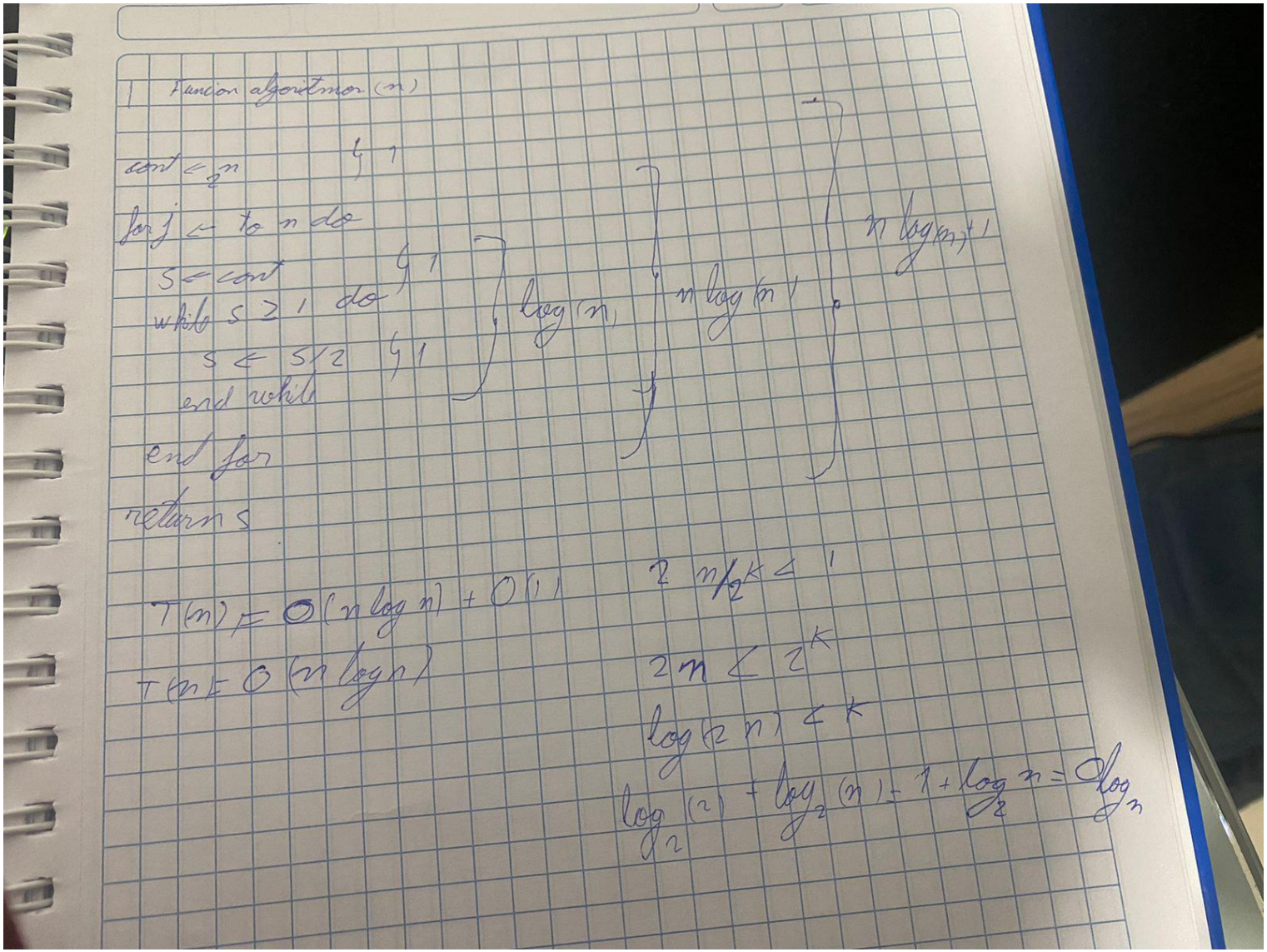
$Tn = O(n\log n) + O(n)$

$Tn = O(n\log n)$

$n/2^k = 1$

$n = 2^k$

$\log_2 n = k$



Function algoritmo (n)

cont ← 2^n { 1

for j ← to n do { $n \log(n)+1$

s ← cont { 1

while s ≥ 1 do { $\log(n)$

s ← s/2 { 1

end while }

end for }

return s

$T(n) = O(n \log n) + O(1)$

$T(n) = O(n \log n)$

$2\ n/2^k < 1$

$2n < 2^k$

$\log(2n) < k$

$\log_2(2) + \log_2(n) = 1 + \log_2 n = O\log_n$