Q 2.1a answer) t1(n) ∈ O(n2) and t2(n) ∈ O(n3)

**Proof for t1(n) ∈ O(n2):**

Defn : O(g) = {f |∃k ∈ N.f ≤a k · g}

To prove:

t1(n) ∈ O(n2)

From definition:

∃ k ∈ N and n0 ∈ N such that t1(n) ≤ kn2 for n > n0.

Taking k = 6,

So, t1(n) = 5n2 + 16 ≥ 6n2 for n ∈ {0, 1, 2, 3, 4}, but t1(n) ≤ 6n2 for n > 4 = n0.

Hence, t1(n) is in O(n2).

**Proof for t1(n) ∈ O(n2):**

Defn : O(g) = {f |∃k ∈ N.f ≤a k · g}

To prove:

t2(n) ∈ O(n3)

From definition:

∃ k ∈ N and n0 ∈ N such that t1(n) ≤ kn2 for n > n0.

Taking k = 7,

So, t2(n) = 6n3+n2+18 ≥ 7n3 for n ∈ {0, 1, 2, 3}, but t2(n) ≤ 7n3 for n > 4 = n0.

Hence, t2(n) is in O(n3).

Q 2.1b answer) As the program runs sequentially the total time is just the addition of the time taken by t1(n)and t2, the program belongs to the time complexity for the function t1(n) + t2(n).

So, t1(n) + t2(n) ∈ O(n3)

**Proof for t1(n) + t2(n) ∈ O(n3):**

Defn : O(g) = {f |∃k ∈ N.f ≤a k · g}

To prove:

t1(n) + t2(n) ∈ O(n3)

From definition:

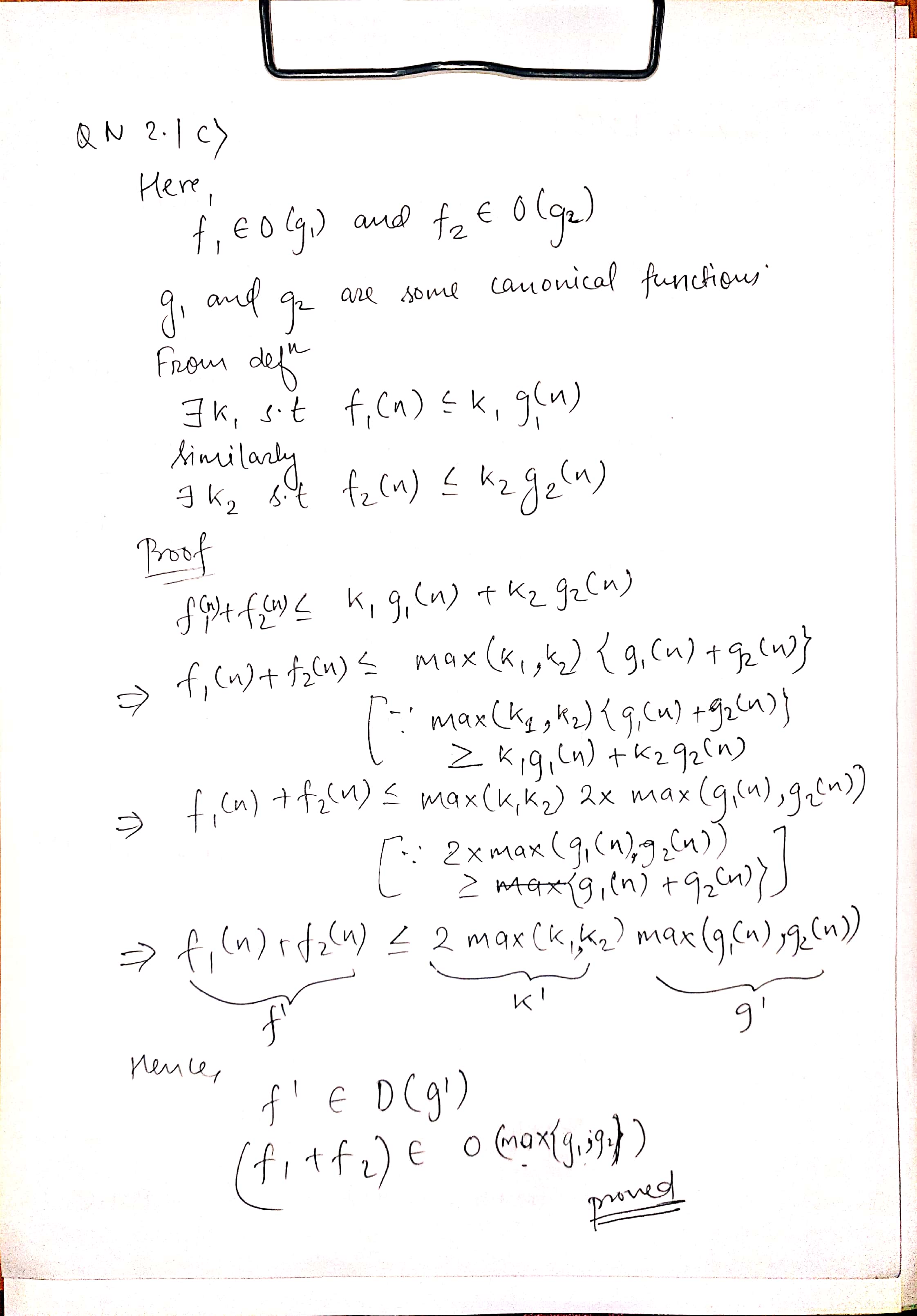
∃ k ∈ N and n0 ∈ N such that t1(n)+t2(n) ≤ kn2 for n > n0.

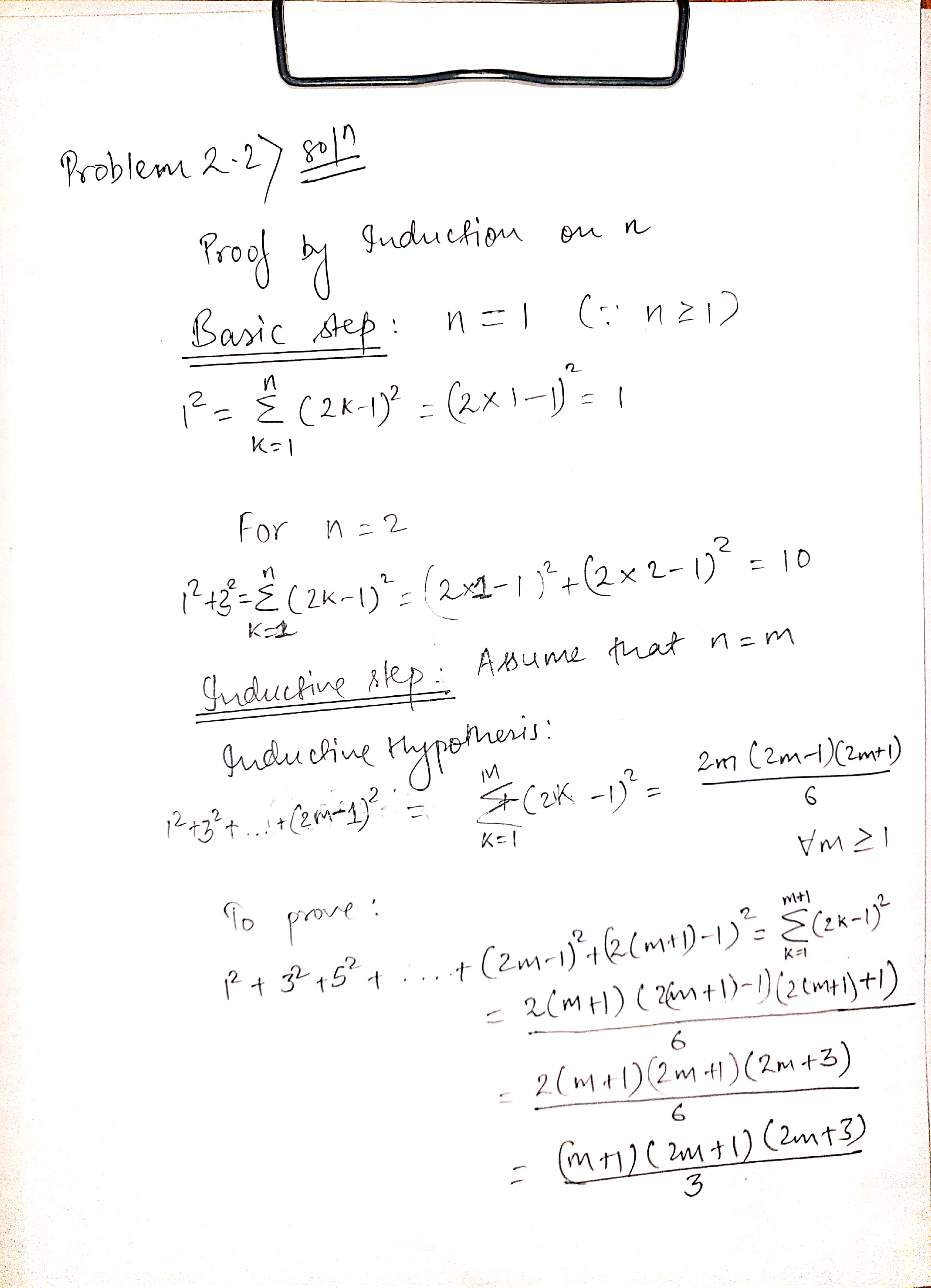
t1(n) + t2(n) = 6n3+6n2+34

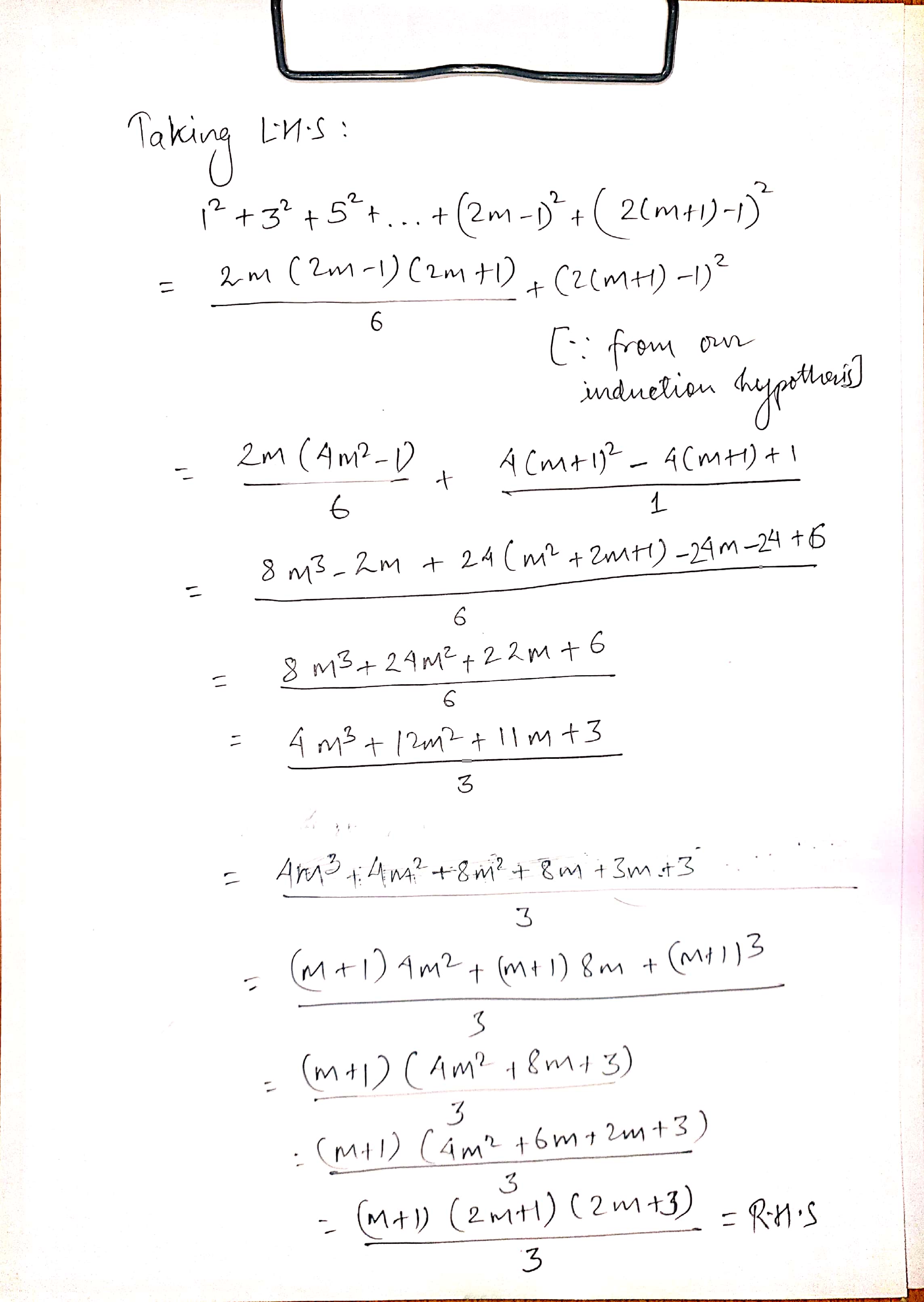
Taking k = 7,

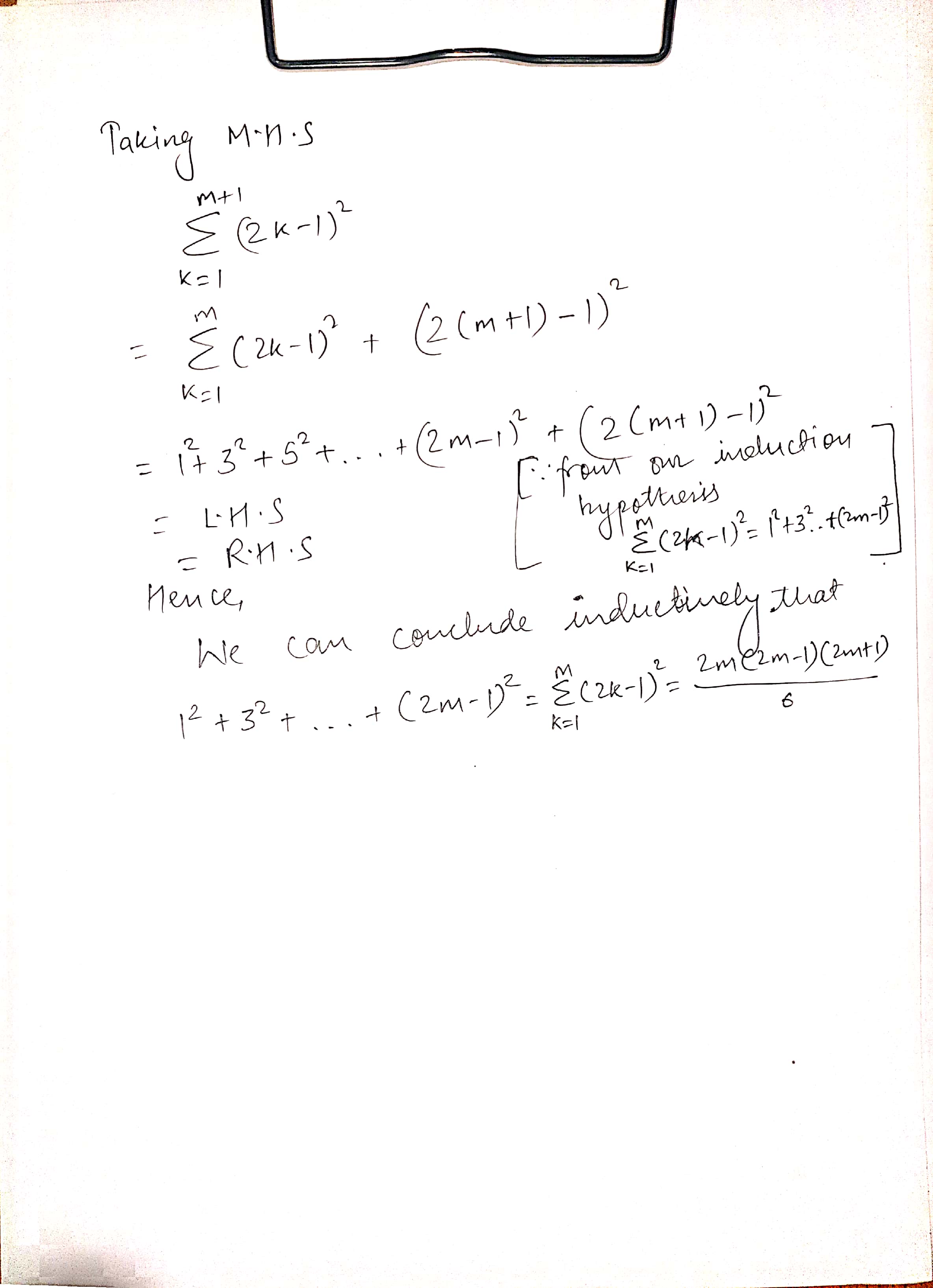
So, t1(n) + t2(n) = 6n3+6n2+34 ≥ 7n3 for n ∈ {0, 1,…6}, but t1(n)+t2(n)≤ 7n3 for n > 6 = n0.

Hence, t1(n) + t2(n) is in O(n3).









Problem 2.3 solution

(a)

factors :: Integral a => a -> [a]

factors n = [x | x <- [1..n], (n mod x) == 0]

(b)

pythagoras :: (Num a, Eq a, Enum a) => a -> [(a,a,a)]

pythagoras x = [(a,b,c) | a <- [1..x], b <- [1..a], c <- [1..x], (a)^2 + (b)^2 == (c)^2]