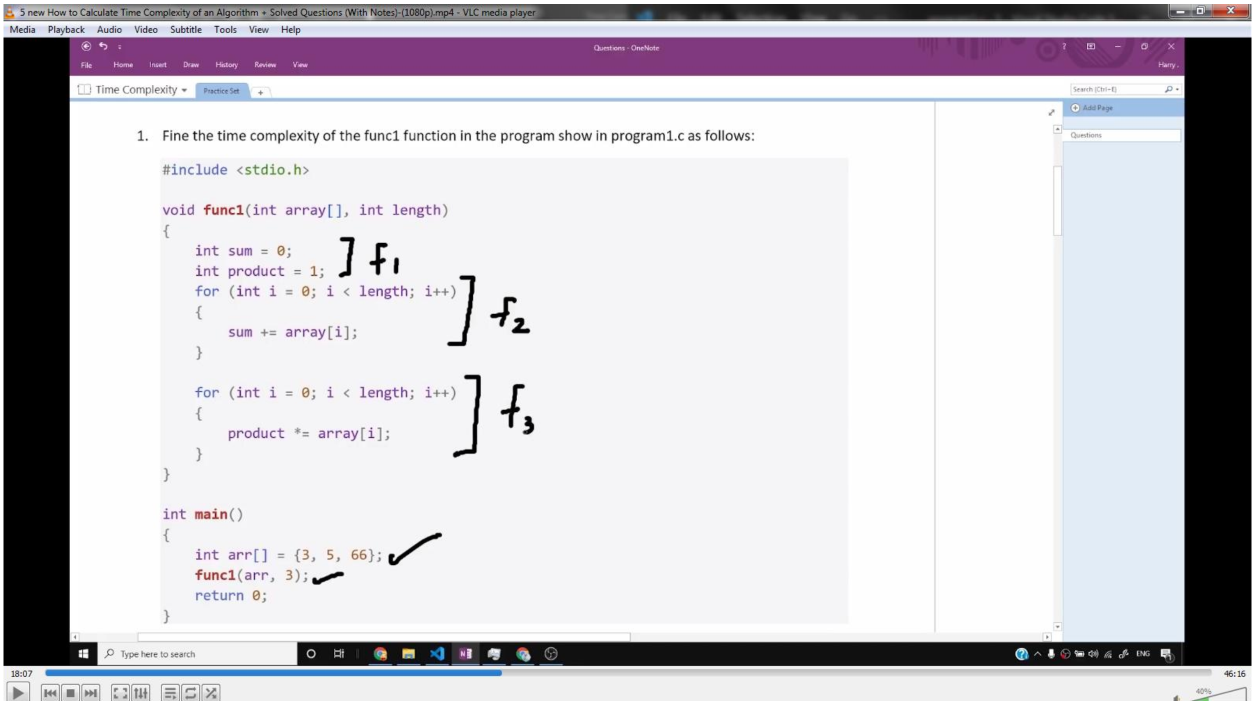
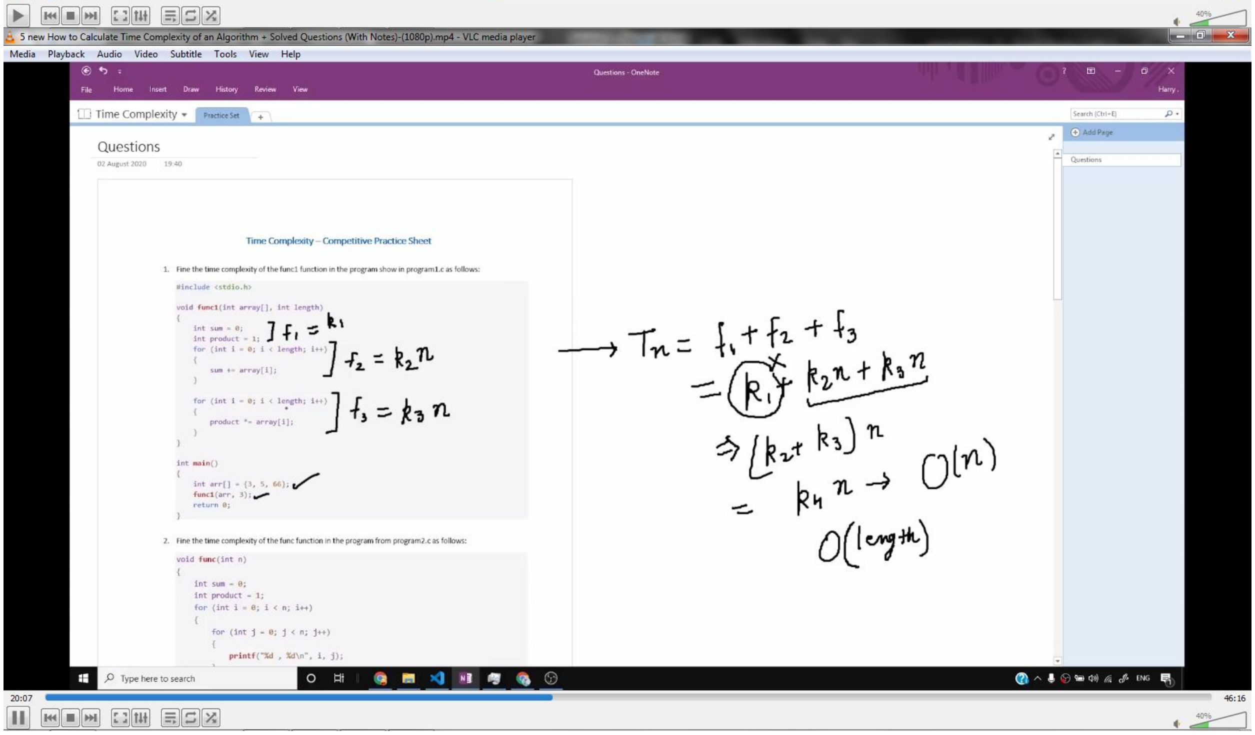
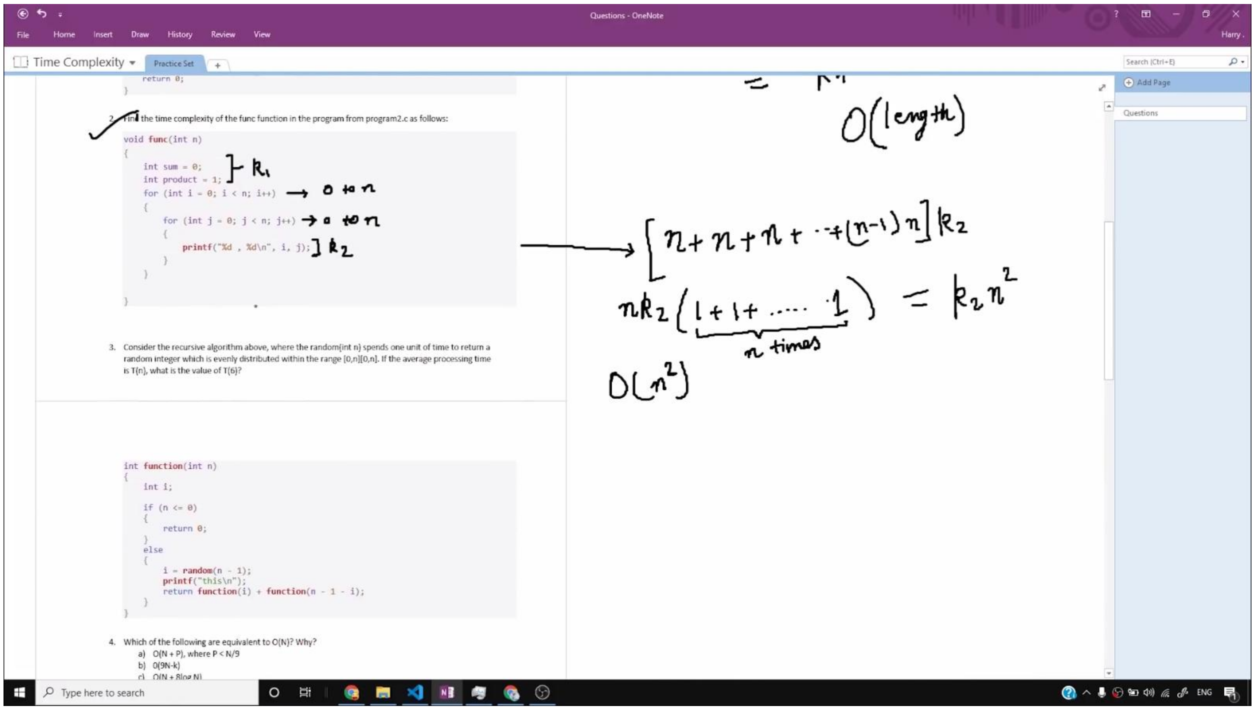
Solution 1:

Divide the program in fragments





Solution 2:



Questions - OneNote
File
Home
Insert
Draw
History
Review
View
Time Complexity
Practice Set
2. Find the time complexity of the func function in the program from program2.c as follows:
void func(int n)
{
int sum = 0;
int product = 1;
k1
for (int i = 0; i < n; i++)
0 to n
{
for (int j = 0; j < n; j++)
0 to n
{
printf("%d , %d\n", i, j);
k2
}
}
}
3. Consider the recursive algorithm above, where the random(int n) spends one unit of time to return a random integer which is evenly distributed within the range [0,n][0,n]. If the average processing time is T(n), what is the value of T(6)?
int function(int n)
{
int i;
if (n <= 0)
{
return 0;
}
else
{
i = random(n - 1);
printf("this\n");
return function(i) + function(n - 1 - i);
}
}
4. Which of the following are equivalent to O(N)? Why?
a) O(N + P), where P < N/9
b) O(9N-k)
O(length)
[n + n + n + ... + (n-1) n] k2
n k2 (1 + 1 + ... 1) = k2 n^2
n times
O(n^2)
Add Page
Questions
Type here to search

Solution 3:

Questions - OneNote

File Home Insert Draw History Review View

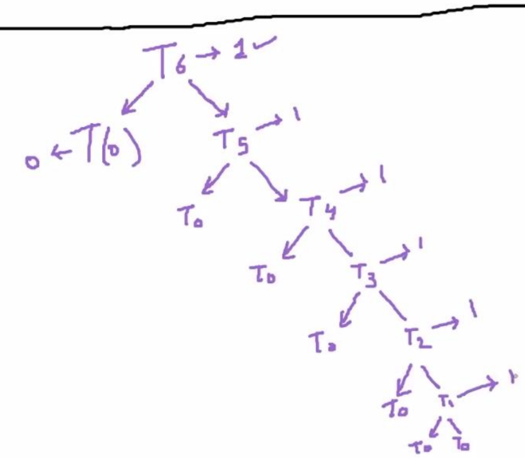
Time Complexity Practice Set

...is evenly distributed within the range [0,n]~~[0,n)~~. If the average processing time ...lue of T(6)? (—)

random(6) → [0, 6]

[0, 5]

$O(n^2)$




```
n)
→ k₁ = 0
;
om(n - 1);  → 1
this\n");
unction(i) + function(n - 1 - i);
```

...g are equivalent to O(N)? Why?
...re P < N/9

...code sums the values of all the nodes in a balanced binary search tree. What is its

```
node)
== NULL)
n 0;
```

Type here to search

Questions - OneNote

File Home Insert Draw History Review View

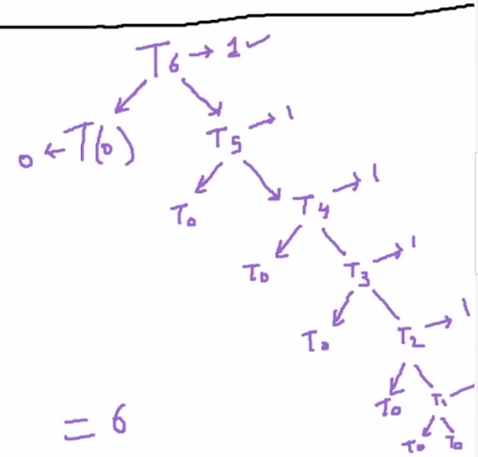
Time Complexity Practice Set

random integer which is evenly distributed within the range [0,n]~~[0,n)~~. If the average processing time is T(n), what is the value of T(6)? (—)

random(6) → [0, 6]

[0, 5]

$O(n^2)$



= 6

```
int function(int n)
{
    int i; ] → k₁ = 0

    if (n <= 0)
    {
        return 0;
    }
    else
    {
        i = random(n - 1);  → 1
        printf("this\n");
        return function(i) + function(n - 1 - i);
    }
}
```

4. Which of the following are equivalent to O(N)? Why?
   a) O(N + P), where P < N/9
   b) 0(9N-k)
   c) O(N + 8log N)
   d) $O(N + M^2)$

5. The following simple code sums the values of all the nodes in a balanced binary search tree. What is its runtime?

```
int sum(Node node)
{
    if (node == NULL)
    {
        return 0;
    }
```

Type here to search

Questions - OneNote

File Home Insert Draw History Review View

Time Complexity | Practice Set

```
        }
    }
}
```

3. ~~Consider~~ Consider the recursive algorithm ~~above~~ below, where the random(int n) spends one unit of time to return a random integer which is evenly distributed within the range [0,n]~~[0,n)~~. If the average processing time is T(n), what is the value of T(6)? (—)

random(6) → [0, 6]

[0, 5]

```
int function(int n)
{
    int i;        ] → k1 = 0

    if (n <= 0)
    {
        return 0;
    }
    else
    {
        i = random(n - 1);   → 1
        printf("this\n");
        return function(i) + function(n - 1 - i);
    }
}
```

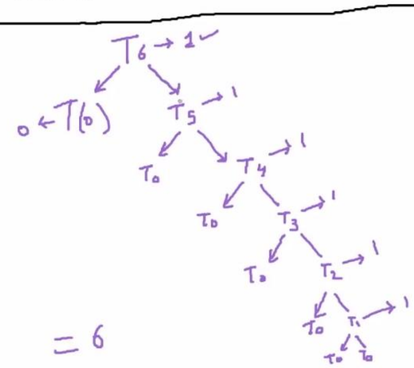
4. Which of the following are equivalent to O(N)? Why?
    a) O(N + P), where P < N/9
    b) O(9N-k)
    c) O(N + 8log N)
    d) O(N + M²)

5. The following simple code sums the values of all the nodes in a balanced binary search tree. What is its runtime?

```
int sum(Node node)
{
    if (node == NULL)
    {
        return 0;
    }
    return sum(node.left) + node.value + sum(node.right);
}
```

$$nk_2(\underbrace{1+1+\cdots 1}_{n \text{ times}}) = k_2 n^2$$

$$O(n^2)$$

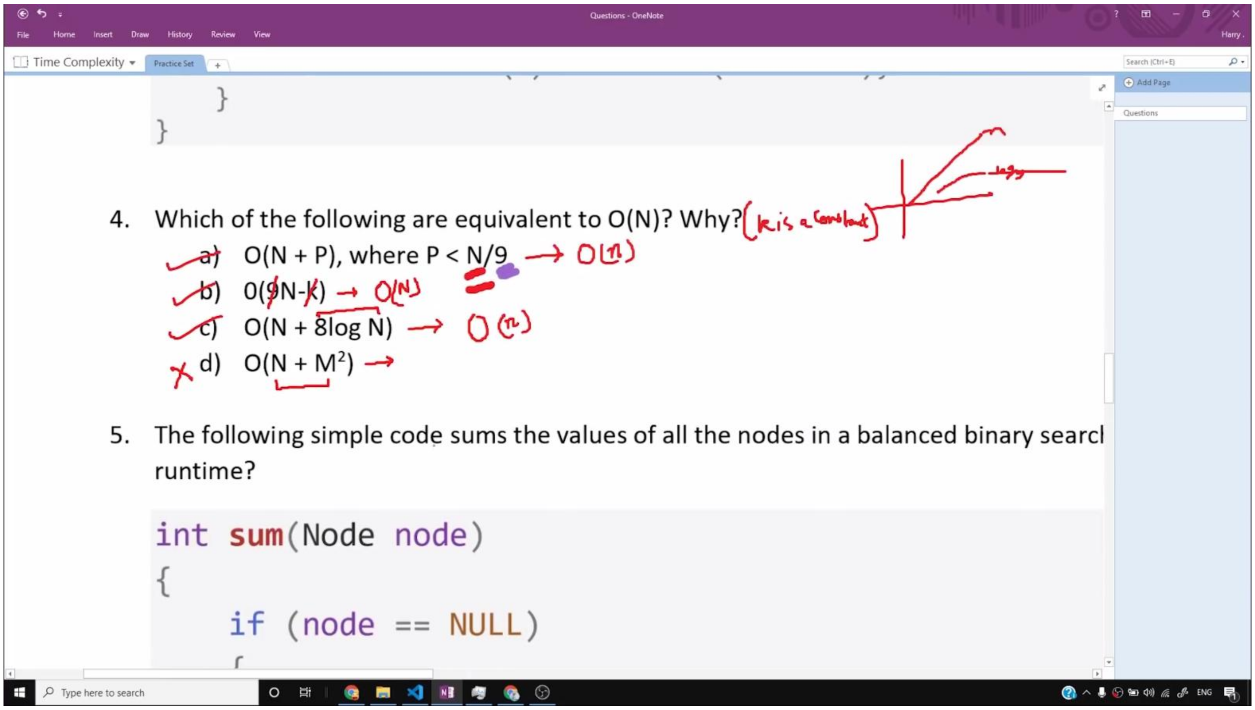


$= 6$

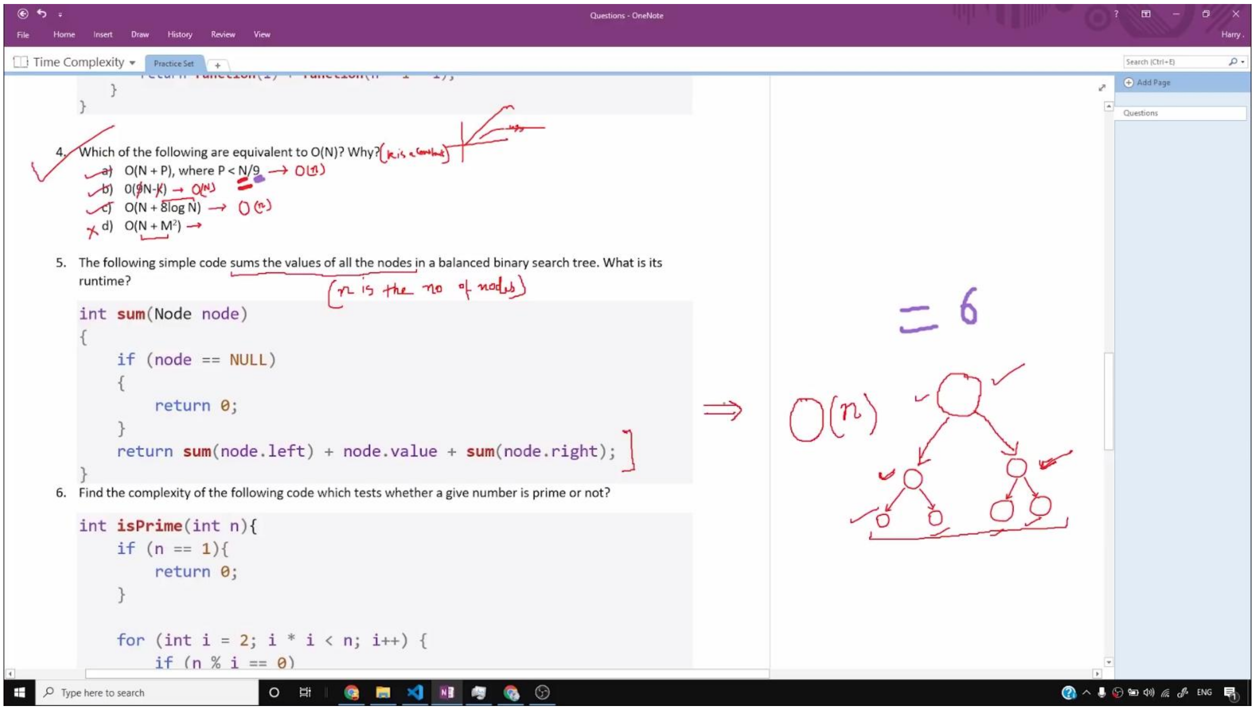
Type here to search

Solution 4:

Here M is variable not a constant

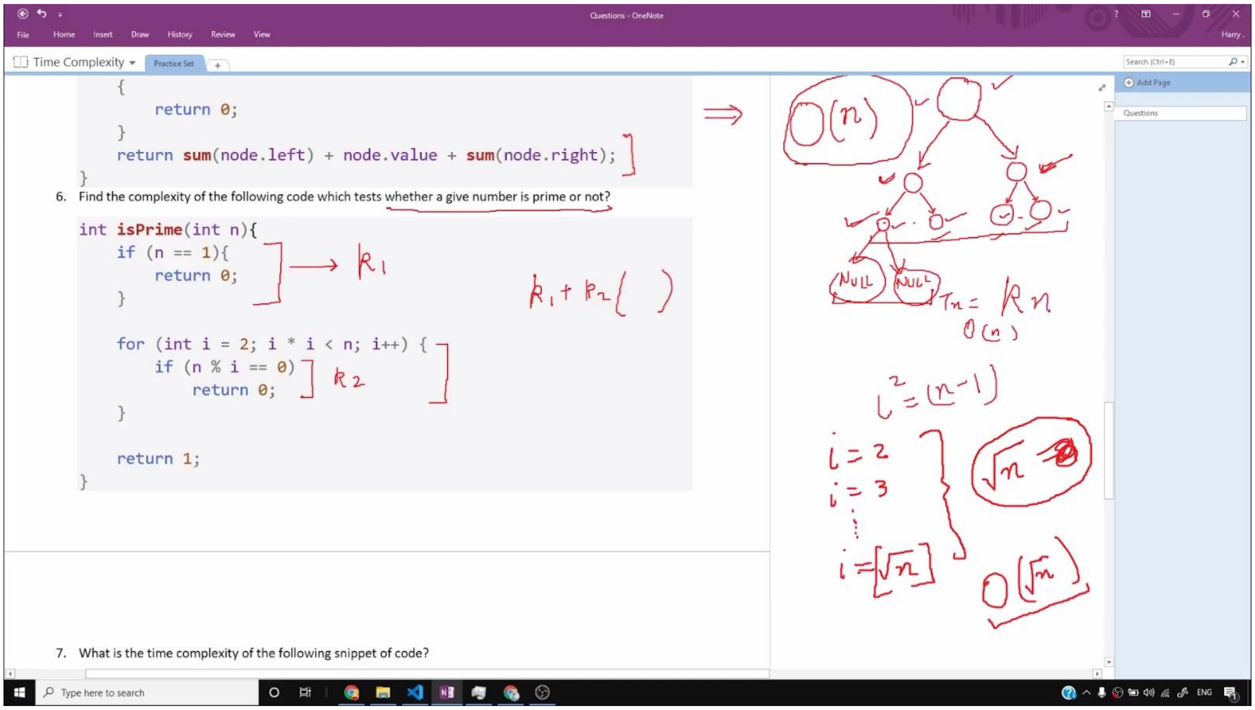


Solution 5:

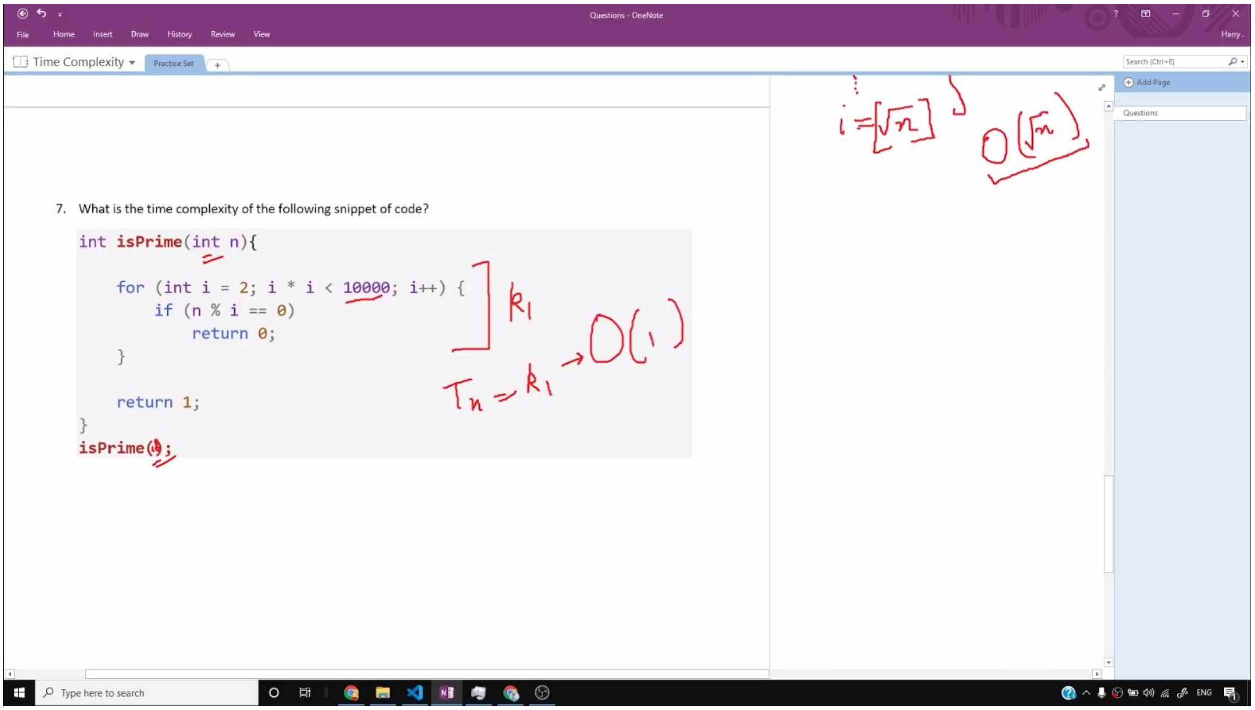


Questions - OneNote
File
Home
Insert
Draw
History
Review
View
Time Complexity
Practice Set
Add Page
Questions
4. Which of the following are equivalent to O(N)? Why? (k is a constant)
a) O(N + P), where P < N/9 → O(N)
b) O(9N-k) → O(N)
c) O(N + 8log N) → O(n)
d) O(N + M²) →
5. The following simple code sums the values of all the nodes in a balanced binary search tree. What is its runtime?
(n is the no of nodes)
int sum(Node node)
{
if (node == NULL)
{
return 0;
}
return sum(node.left) + node.value + sum(node.right);
}
⟹ O(n)
= 6
6. Find the complexity of the following code which tests whether a give number is prime or not?
int isPrime(int n){
if (n == 1){
return 0;
}
for (int i = 2; i * i < n; i++) {
if (n % i == 0)
Type here to search
ENG

Solution 6:


Questions - OneNote
Time Complexity
Practice Set
return 0;
return sum(node.left) + node.value + sum(node.right);
6. Find the complexity of the following code which tests whether a give number is prime or not?
int isPrime(int n){
if (n == 1){
return 0;
}
for (int i = 2; i * i < n; i++) {
if (n % i == 0)
return 0;
}
return 1;
}
7. What is the time complexity of the following snippet of code?
K1
K2
K1 + K2( )
O(n)
NULL
Tn = Kn
O(n)
i² = (n-1)
i = 2
i = 3
i = √n
√n
O(√n)

Solution 7:



It is running in constant time because “for” loop doesn’t depend on n. It always run for 10000 times i.e. const. So it is O(1)