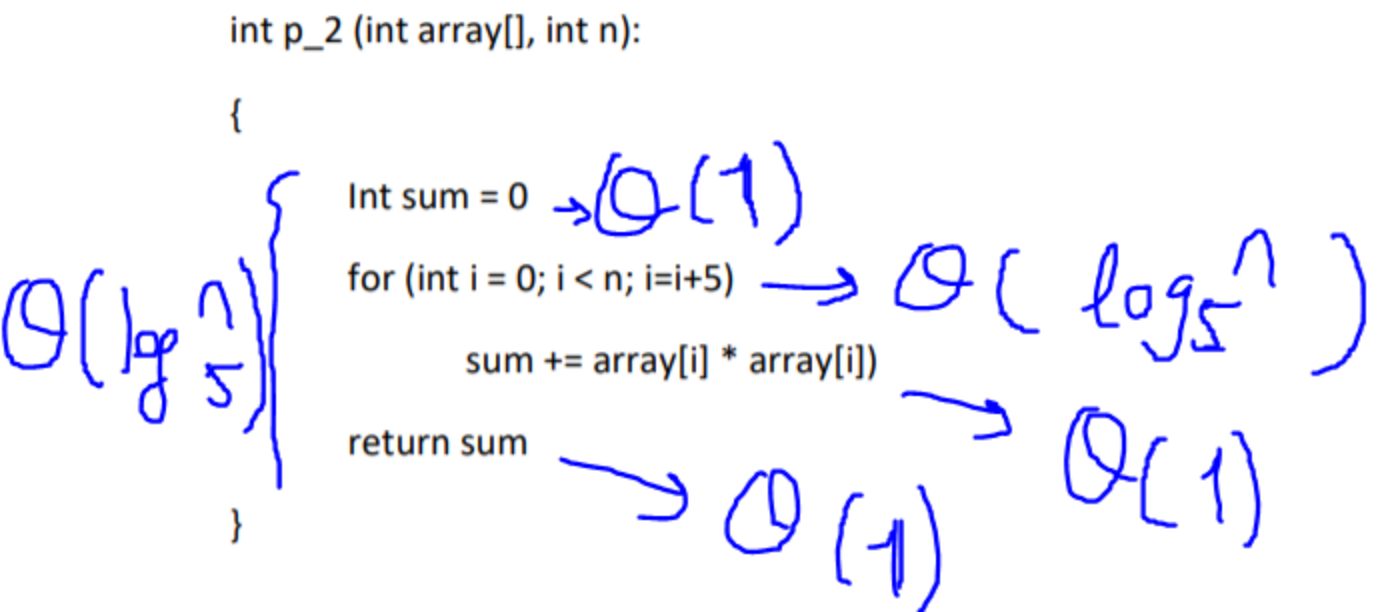
```
int p_1 (int array[]):
{
        return array[0] * array[2])
}
```

Θ(1)

Θ(1)

Θ(1)

```
int p_2 (int array[], int n):
{
    Int sum = 0
    for (int i = 0; i < n; i=i+5)
        sum += array[i] * array[i])
    return sum
}
```



```
void p_3 (int array[], int n):
{
    for (int i = 0; i < n; i++)
        for (int j = 0; j < i; j=j*2)
            printf("%d", array[i] * array[j])
}
```

$\Theta(n)$

$\Theta(\log_2 n)$

$\Theta(n)$

$\Theta(n \cdot \log_2 n)$

void p_4 (int array[], int n):

$O(\log_5 n)$

{

If (p_2(array, n)) > 1000)

$O(\log_5 n)$

p_3(array, n) → $O(n \cdot \log_2 n)$

else

$O(\log_5 n)$

printf("%d", p_1(array) * p_2(array, n)) → $O(\log_5 n)$

$O(1)$ $O(\log_5 n)$

}

+

$O(n \cdot \log_2 n)$

```
int find_min(ArrayList<Integer> arr){
    Integer minimum = arr.get(0) /* Theta(1) */

    for i = 0 to i = n , do; /* Theta(n) */          /* This alghoritm's time complexity is Theta(n) */
        if < arr.get(i) then, /* Theta(1) */
            minimum = arr.get(i) /* Theta(1) */
        end if
}

void is_sum_equal(ArrayList<Integer> arr , int value){

    for i = 0 to i = n, do                          /* Worst case = O(n) , Best case = Theta(1) */
        for j = 0 to j = n, do                      /* Worst case = O(n) , Best case = Theta(1) */

            if ( arr.get(i) + arr.get(j) == value)  /* Theta(1) */
                print arr.get(i) and arr.get(j)     /* Theta(1) */

                return                              /* Theta(1) */
            end if
}
/* Worst case is O(n) * O(n) = O(n^2) and best case is Theta(1) * Theta(1) = Theta(1) */

```