W1D1

Learn and Repeat

Pattern 1. "for all" and "there exists" pattern

Given an array A, if all elements of A are positive, return 1. Else return 0.

Algorithm 1. (Counting method)

```
count = 0;
for (i  = 0; i < n; i++)
        if (A[i] > 0) count++;
(count == n)? true : false;
```

check the condition. If true, increment the counter. If the counter value is n, return true. Else return false.

Algorithm 2. (Test the opposite condition method)

```
for (i  = 0; i < n; i++)
        if (A[i]  <=  0) return false;
return true;
```

check the "opposite condition". If true, return false. Return true outside the loop.

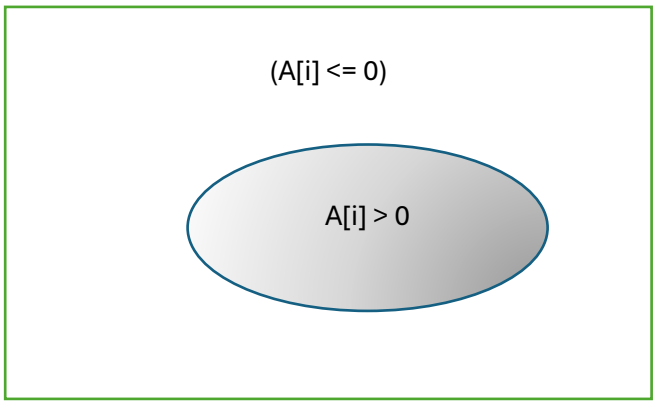
	Best case	Average case	Worst case
Algorithm 1	O(n)	O(n)	O(n)
Algorithm 2	O(1)	O(n)	O(n)

Lower bound of the problem

What is the minimum information required to conclude the problem statement in the affirmative? (This a very confusing idea for students with less exposure to logic.)

You need to check all elements of A. Hence the lower bound of the problem is $\Omega(n)$.

Since the lower bound of the problem is equal to the **worst case time complexity** in Solution 1 (Solution 2), Solution 1 (Solution 2) is optimal.



Exercise 1

An array is defined to be a **235 array** if the number of elements divisible by 2 plus the number of elements divisible by 3 plus the number of elements divisible by 5 plus the number of elements not divisible by 2, 3, or 5 is equal to the number of elements of the array.

Write a method named is123Array that returns 1 if its array argument is a 235 array, otherwise it returns 0.

Note : A number can be divisible by more than one number. For example, 10 is divisible by both 2 and 5.

Algorithm 3. (Counting method)

235Array(A)

count=0;

For each element x in A do

If x mod 2 = 0 then count<- count+1

If x mod 3= 0 then count<- count+1

If x mod 5 = 0 then count<- count+1

If x mod 2 not = 0 AND If x mod 3 not = 0 AND If x mod 5 not = 0 then

count<- count+1

if count = length(A) then return 1

else return 0

Algorithm 4. (Test the opposite condition method)

For i in input:

If (i % 6==0 || i % 10 ==0 || i % 15 ==0) return 0;

End for

Return 1;

Fill in the cells.

	Best case	Average case	Worst case
Algorithm 1	O(n)	O(n)	O(n)
Algorithm 2	O(1)	O(n)	O(n)

What is the lower bound of the problem?

Algo 3 - O(n)

Algo 4 - O(n)