Solved problems for time complexity of loops

Last updated 9/17/2024

General comments/hints

1. If a loop variable takes consecutive values, you do not have write it as a function of the iteration number, r.
2. r can start from 0 or 1, whatever is more convenient in writing the loop variable as a function of r
3. lg = $\log_2$

A) 6.

```
// int mistery(int len, int v); has Θ(len²)
for(i=N; i>=0; i--){
    int res = mistery(N, i);    // ➔ Θ(N²) or O(N²)
    for(k=N; k>=1; k=k-1)
        printf("B");    // ➔ Θ(1) or O(1)
}
```

for-k $TC_{1iter}(k) = O(1)+O(1)+O(1)$

	K		$TC_{1iter}(k) = O(1)$
	N		1
	N-1		1
	N-2		1
	K		1
	1		1
$TC_{fork} = 1+1+...+1 = rows*1 = N * 1 = O(N)$			

for-i: $TC_{1iter}(i) = O(1)+O(N^2)+O(N)+O(1) = O(N^2)$

	i		$TC_{1iter}(i) = O(i^2)$
	N		N^2
	N-1		N^2
	N-2		N^2
	i		N^2
	1		N^2
$TC_{fori} = N^2 + N^2 + ... + N^2 + ... + N^2 = rows * N^2 = N * N^2 = O(N^3)$			

Final answer: $O(N^3)$

A) 7.

```
// int mistery(int N, int v); has O(N²)
for(i=N; i>=0; i--){
     int res = mistery(i, i);   // ➔ O(i²)
     for(k=N; k>=1; k=k-1)
          printf("B");
}
```

for-k $TC_{1iter}(k) = O(1)+O(1)+O(1)$

	k		$TC_{1iter}(k) = O(1)$
	N		1
	N-1		1
	N-2		1
	k		1
	1		1
$TC_{fork} = 1+1+...+1 = rows*1 = N * 1 = O(N)$			

for-i: $TC_{1iter}(i) = O(1)+O(i^2)+O(N)+O(1) = O(i^2)$

	i		$TC_{1iter}(i) = O(i^2)$
	N		N^2
	N-1		$(N-1)^2$
	N-2		$(N-2)^2$
	i		i^2
	1		1^2
$TC_{fori} = 1^2 + 2^2 + ... + i^2 + ... + (N-2)^2 + (N-1)^2 + N^2 = N(N+1)(2N+1)/2 = (2N^3 + 3N^2 + N)/2 = O(N^3)$			

Final answer: $O(N^3)$

A) 8

```
for (i = 101; i<=(100+N); i++)
   for (k=1; k<=i; k = k+1)
      printf("B ");
```

for-k $TC_{1iter}(k) = O(1)+O(1)+O(1)$

	k		$TC_{1iter}(k) = O(1)$
	i		1
	i-1		1
	i-2		1
	k		1
	1		1
$TC_{fork} = 1+1+...+1 = rows*1 = i*1 = O(i)$			

for-i: $TC_{1iter}(i) = O(1)+O(i)+O(1) = O(i)$

r	i	i = fct(r)	$TC_{1iter}(i) = O(i)$
1	101	=100+1	101
2	102	=100+2	102
3	103	=100+3	103
r	i	=100+r	i
r_{last}	100+N	$=100+r_{last}$	100+N
TC_{fori} = 101 + 102 + 103 + ... + i + ... (100+N)= (100+1)+(100+2)+(100+3)+...+(100+r)+ ... +(100+N) =(100+100+...+100+...+100) + (1+2+3+..+r+...N) = =100*rows + N(N+1)/2 = 100*N + $(N^2+N)/2 = O(N^2)$			

Final answer: $O(N^2)$