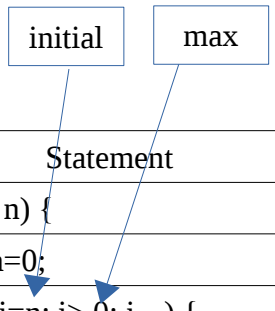

	Statement	S/E	Frequency	Total
1	int func(int n) {	0	-	0
2	int sum=0;	1	1	1
3	for(int i=n; i> 0; i- -) {	1	n+1	n+1
4	for(int j=n-1; j>=i; j- -) {	1	n(n + 1)/2	n(n + 1)/2
5	sum=i+j;	1	n(n − 1)/2	n(n − 1)/2
6	System.out.println(sum);	1	n(n − 1)/2	n(n − 1)/2
7	}	0	-	0
8	}	0	-	0
9	}	0	-	0
Total Operation			$\frac{3}{2}n^2+\frac{1}{2}n+2$	
Big-Oh			$O(n^2)$	

1. This line is not executable, S/E is 0.
2. This line is executable (S/E is 1) and it will be executed once (Frequency = 1)
3. This line is executable (S/E is 1) and it will be executed n+1 times, we applied the rule for the *for* statement initial – max + 1 = n – 0 + 1 = n + 1
4. int this step the inside for depends in I, see the following table:

i	j (initial)	How many times j >= i	sum=i+j	System.out.println(sum);
n	n-1	n-1 >= n (1 time)	0	0
n-1	n-1	n-1 >= n-1 , n-2 >= n-1 (2 times)	1	1
n-2	n-1	n-1 >= n-1 , n-2 >= n-1 …. (3 times)	2	2
n-3	n-1	n-1 >= n-1 , n-2 >= n-1 …. (4 times)	3	3
….	….	….	….	….
3	n-1	n-1 >= n-1 , n-2 >= n-1 …. (n-2 times)	n-3	n-3
2	n-1	n-1 >= n-1 , n-2 >= n-1 …. (n-1 times)	n-2	n-2
1	n-1	n-1 >= n-1 , n-2 >= n-1 …. (n times)	n-1	n-1
0	….	….	….	….

j >= i = 1 + 2 + 3 + 4 + …. + (n-2) + (n-1) + (n) = $\sum_{i=1}^{n} i=\frac{n(n+1)}{2}$

sum=i+j = 0 + 1 + 2 + 3 + …. + (n-3) + (n-2) + (n-1)

= $\sum_{i=1}^{n} i=\frac{n(n+1)}{2}-n=\frac{n^2+n}{2}-n=\frac{n^2+n}{2}-\frac{n}{1}=\frac{n^2+n-2n}{2}=\frac{n^2-n}{2}=\frac{n(n-1)}{2}$