Counting Operations

On this worksheet, we will only count 'basic units': assignments statements (A) and comparisons (C).

1) Count the number of basic operations for the worst-case:

$a \leftarrow 3 * n$	1A
$count \leftarrow 0$	1A
while $a > n$ **do**	2n(1C) + 1C
$a \leftarrow a - 1$	2n(1A)
$count \leftarrow count + 1$	2n(1A)
end	T(n) = 2 + 2n(3) + 1 = 6n + 3

2) Count the number of basic operations for the worst-case:

$s \leftarrow 0$	1A
for $k \leftarrow 1$ **to** n **do**	1A + n(1C + 1A) + 1C
$s \leftarrow s + (k * k)$	n(1A)
end	T(n) = 2 + n(3) + 1 = 3n + 3

3) Count the number of basic operations for the worst-case:

Algorithm find(A, n, key):
    Input: An array A storing $n \geq 1$ integers; an integer key to search for
    Output: The index of key in A

$k \leftarrow 0$	1A
while $k < n$ **do**	n(1C + 1C + 1A) + 1C
if $A[k] = key$ **do**	
return k	T(n) = 1 + n(3) + 1 + 1 = 3n + 3
end	
$k \leftarrow k + 1$	
end	
return "not found"	1A

4) Count the number of basic operations for the worst-case:

for $k \leftarrow 0$ **to** $n - 1$ **do**	1A + n(1C + 1A) + 1C
for $j \leftarrow 0$ **to** $n - 1$ **do**	n (1A + n(1C + 1A + 1C + 3A) + 1C)
if $A[k] \leq A[j]$ **then**	
swap(A, j, k)	
end	T(n) = 2+2n+2n+6n^2= 6n^2 + 4n + 2
end	
end	

5) Count the number of basic operations for the worst-case:

$s \leftarrow 0$	1A
for $i \leftarrow 1$ **to** n **do**	1A + n(1C + 1A) + 1C
for $j \leftarrow 1$ **to** i **do**	n (1A + ??(1C + 1A + 2A) + 1C) we can re-write as:
$s \leftarrow s + i$	
$s \leftarrow s + i$	n (1A + 1C) + X (1C + 1A + 2A)
end	
end	

T(n) = 2+n(4) + $\frac{n(n+1)}{2}$(4) + 1

T(n) = 3 + 4n + 2n^2+2n = 2n^2 + 6n + 3

How many times are the 4 operations in X executed?

when i is 1: once, when i is 2: twice, …

1 + 2 + 3 + 4 + 5 + … + n-2 + n-1 + n times.

If we add (1+n), (2+n-1), (3+n-2) we have n/2 pairs of (n+1).

So the 4 operations are executed $\frac{n(n+1)}{2}$ times