Dennis Kuzminer
CSCI-UA 310-001 PS1

9.

a. i=0 → vec.append(1) will call resize(1) which will set newsz to 1. Line 13 was executed 0 times.
i=1 → resize(2) will be called which will have line 13 be executed once as size is not yet updated.
i=2 → resize(3) will be called which will have line 13 be executed twice.
i=3 → resize(4) will be called which will have line 13 be executed three times.
It is clear to see that there is a pattern. vec.append is called n times and for every ith iteration, line 13 is called i times. This means that the formula for the number of times line 13 is called is essentially n * n (n(n-1)/2)(number of loops in the outer for-loop * number of loops in the inner for-loop). This means that the number of times line 13 is executed is **Θ(n^2)**.

b. i=0 → vec.append(1) will call resize(1) which will set newsz to 1. Line 13 was executed 0 times.
i=1 → resize(2) will be called which will have line 13 be executed once.
i=2 → resize(3) will be called which will have line 13 be executed twice.
i=3 → resize(4) will be called which will have line 13 not be executed.
i=4 → resize(5) will be called which will have line 13 be executed four times.
When i is a power of 2, only then will the nested loop execute.
$2^{\lfloor log_2(n) \rfloor}$ → Which is essentially **O(n)**

Outer For Loop i=0	Inner For Loop line 13	You can rewrite this to make it so	Outer For Loop i=0	Redistributed series
1	1	that each ‘i’	1	1
2	2	will have	2	1
3	0	only 1 other	3	1
4	4	execution.	4	1
5	0	This makes	5	1
6	0	O(n)	6	1
7	0		7	1
8	8		8	1