BST Worksheet

Suppose that we have numbers between 1 and 1000 in a binary search tree and we want to search for the number 363. Which of the following sequences could not be the sequence of nodes examined?

1. 2, 252, 401, 398, 330, 344, 397, 363.
2. 924, 220, 911, 244, 898, 258, 362, 363.
3. 925, 202, 911, 240, 912, 245, 363.

   In step 3 we go from 911 to 240 which means we take the left path. In the very next step, we hit 912 which cannot be contained in a left subtree of the 911 node.
4. 2, 399, 387, 219, 266, 382, 381, 278, 363.
5. 935, 278, 347, 621, 299, 392, 358, 363.

   After taking the right path from 347 we encounter the value 299 which cannot be contained in a right subtree of the 347 node.

Dynamic Array Analysis Worksheet

BEGIN ALGORITHM

1. IF the actual number of occupied slots in the array is less than the physical capacity of the array, place new element in the leftmost vacant slot and you're done.
2. ELSE
    - (a) Create a new array with twice the capacity of the existing array.
    - (b) Copy elements from existing array to new array.
    - (c) Add element to the leftmost location in the new array
    - (d) Delete old array

END ALGORITHM

Start with an empty array of capacity 1. Now, insert elements 1 through 16 one by one into the array doubling the capacity when needed as described above. Calibrate the run time of this sequence of insertions by solely counting the *number of single-element copy operations* for each insertion. Draw diagrams if needed.

Insertion	# Single-element copies
1	0
2	1
3	2
4	1
5	4
6	0
7	0
8	0
9	8
10	0
11	0

12	0
13	0
14	0
15	0
16	0
TOTAL	16