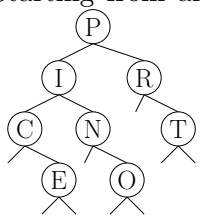
COS 126 Binary Search & Binary Trees (Section 4.4)

1. Starting from an empty binary search tree, we inserted the letters P, R, I, N, C, E, T, O, N and got:



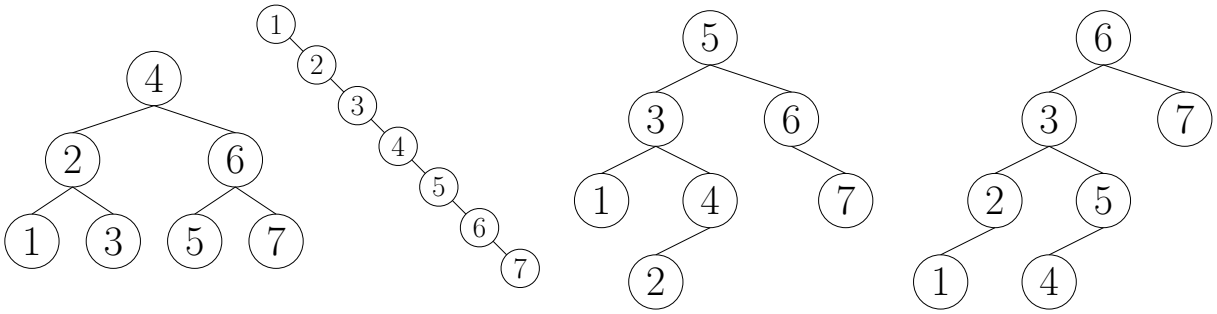
   a. What keys are examined when we search for E? **Ans.: P, I, C, E**
   b. What keys are examined when we search for Q? **Answer: P, R**

2. To insert an item into a binary tree, you (a) search for it, and (b) insert it where the search ended if it was not found. Build a new binary tree, starting from an empty tree and inserting I, N, S, E, R, T, U, S, P, L, Z.

   Answer below; it has been rotated for space reasons. Turn your head 45°. I is the root.

```
I-N-S-T-U-Z
| | |
E L R
    |
    P
```

3. Which of the following is *not* a valid binary search tree? What number cannot be found when we search for it? Of the valid ones, which one leads to the fastest searches?



   Answer: the third one is invalid — 2 cannot be found when we search for it. For performance, it depends on how often each key occurs. *If* all occur with the same probability, the first one is best.

4. (4.4.9) Suppose we have int values between 1 and 1000 in a BST and search for 363. Which of the following *cannot* be the sequence of keys examined?

   (a) 2, 252, 401, 398, 330, 363
   (b) 399, 387, 219, 266, 382, 381, 278, 363
   (c) 3, 923, 220, 911, 244, 898, 258, 362, 363
   (d) 4, 924, 278, 347, 621, 299, 392, 358, 363
   (e) 5, 925, 202, 910, 245, 363

   Answer: (d). 299 cannot appear after 621 since that would place it to the right of 347.