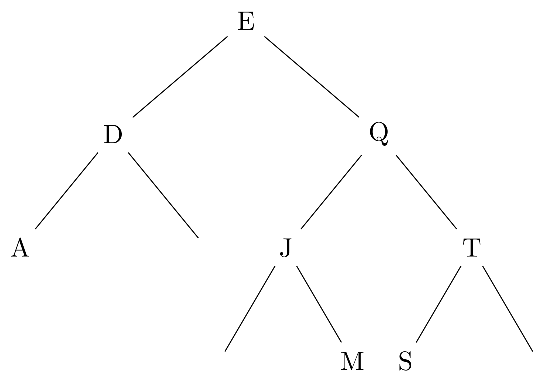
Exercise 1 Simple training exercises

1. For the keys $\{1, 4, 5, 10, 16, 17, 21\}$, draw binary search trees of heights 2, 3, 4, 5 and 6 (CLRS 12.1-1)
2. You are searching for the number 363 in binary search tree containing numbers between 1 and 1000. Which of the following sequences *cannot* be the sequence of nodes examined? (CLRS 12.2-1)
   - 2, 252, 401, 398, 330, 344, 397, 363
   - 924, 220, 911, 244, 898, 258, 362, 363
   - 925, 202, 911, 240, 912, 245, 363
   - 2, 399, 387, 219, 266, 382, 381, 278, 363
   - 935, 278, 347, 621, 299, 392, 358, 363
3. Give the sequence of nodes examined when the following calls are made on the binary search tree given beneath:
   - Tree-Search(T, M)
   - Tree-Minimum(Q)
   - Tree-Successor(M)



4. Write the Tree-Predecessor procedure (CLRS 12.2-3)

Exercise 2 Fun creative exercises!

1. Professor Kilmer claims to have discovered a remarkable property of binary search trees. Suppose that the search for key k in a binary search tree ends up at a leaf. Consider three sets: A, the keys to the left of the search path; B, the keys on the search path; and C, the keys to the right of the search path. Professor Kilmer claims that any three keys $a \in A$, $b \in B$, and $c \in C$ must satisfy $a \leq b \leq c$. Give a smallest possible counterexample to the professor's claim. (CLRS 12.2-4)
2. Show that if a node in a binary search tree has two children, then its successor has no left child and its predecessor has no right child. (CLRS 12.2-5)
3. An alternative method of performing an inorder tree walk of an n-node binary search tree finds the minimum element in the tree by calling Tree-Minumum and then making $n - 1$ calls to Tree-Successor. Prove that this algorithm runs in $\Theta(n)$ time. (CLRS 12.2-7)