[320] Breadth First Search

Tyler Caraza-Harter

Review

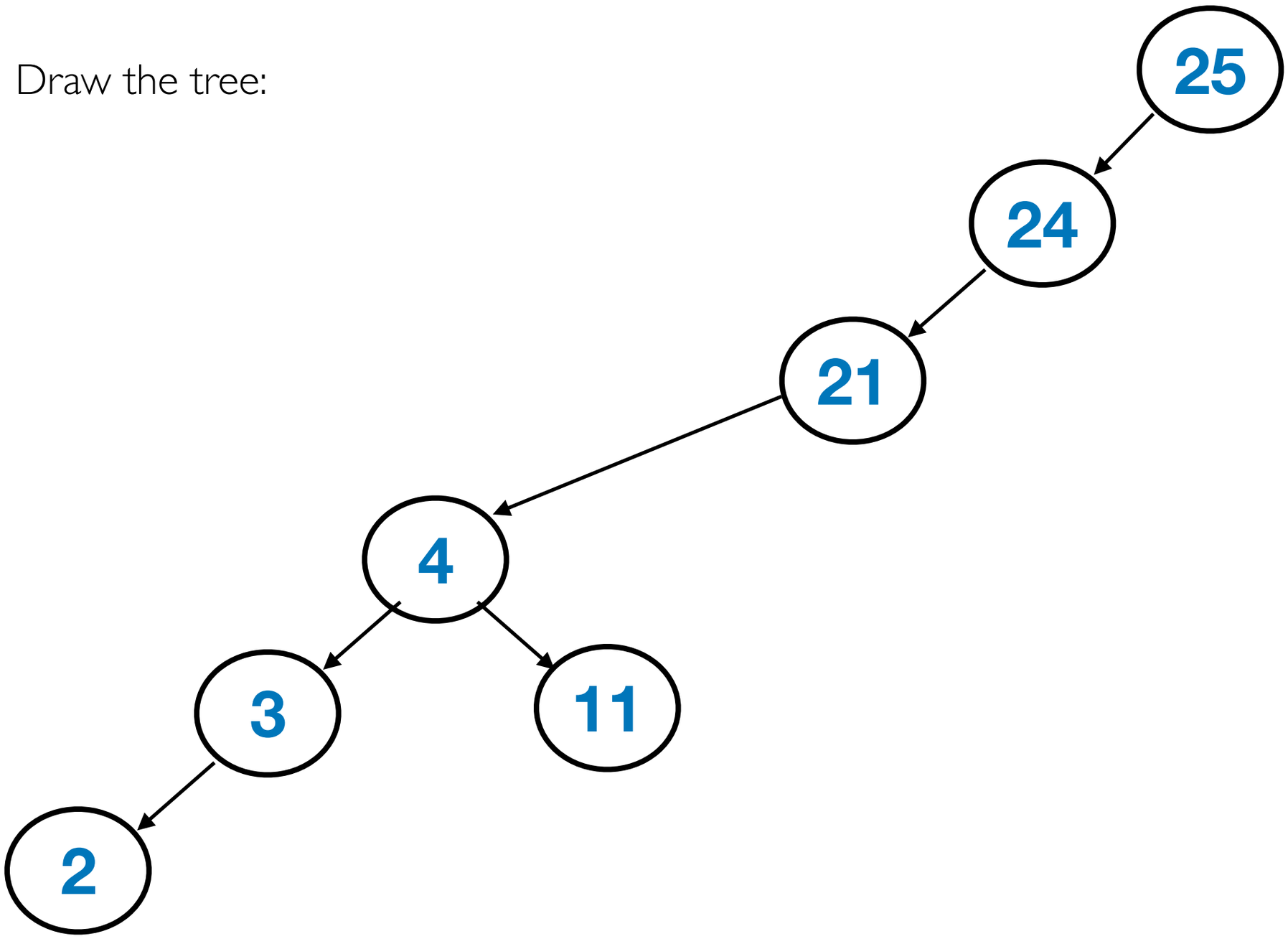
Assume this insertion order for a BST: **25, 24, 21, 4, 3, 2, 11**

Draw the tree:

Review

Assume this insertion order for a BST: **25, 24, 21, 4, 3, 2, 11**

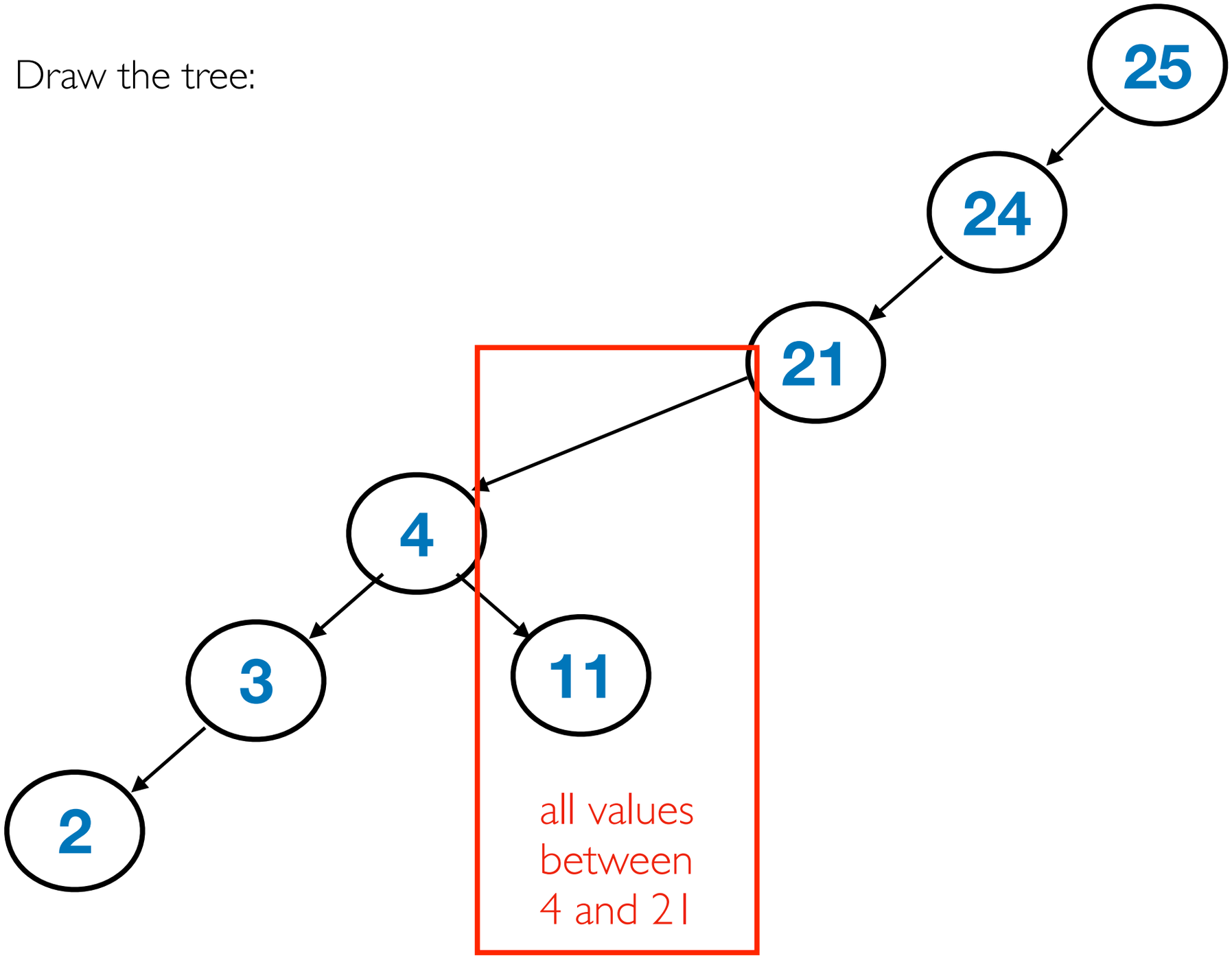
Draw the tree:



Review

Assume this insertion order for a BST: **25, 24, 21, 4, 3, 2, 11**

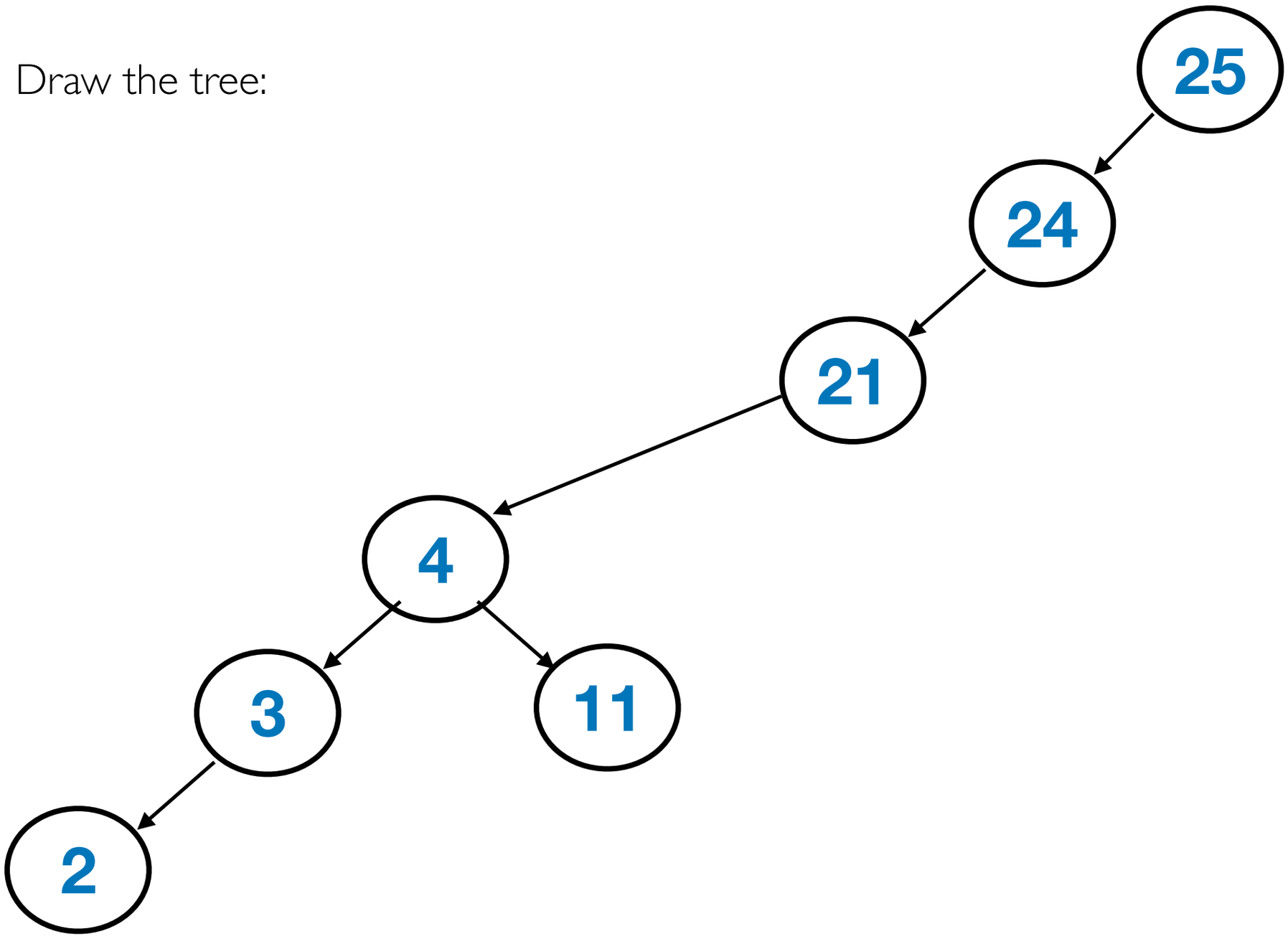
Draw the tree:



Review

Which nodes will be checked if we're searching for 22?

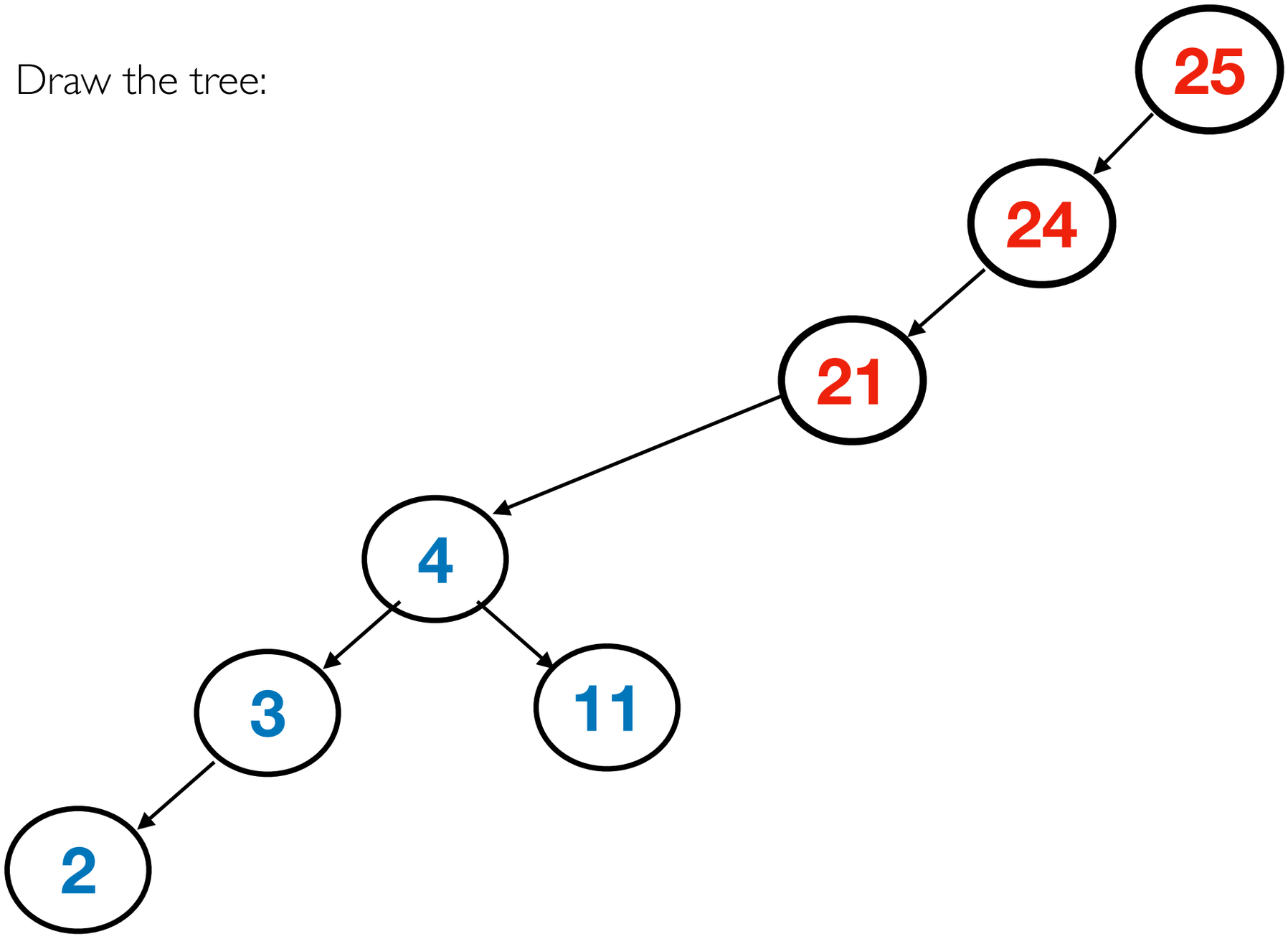
Draw the tree:



Review

Which nodes will be checked if we're searching for 22?

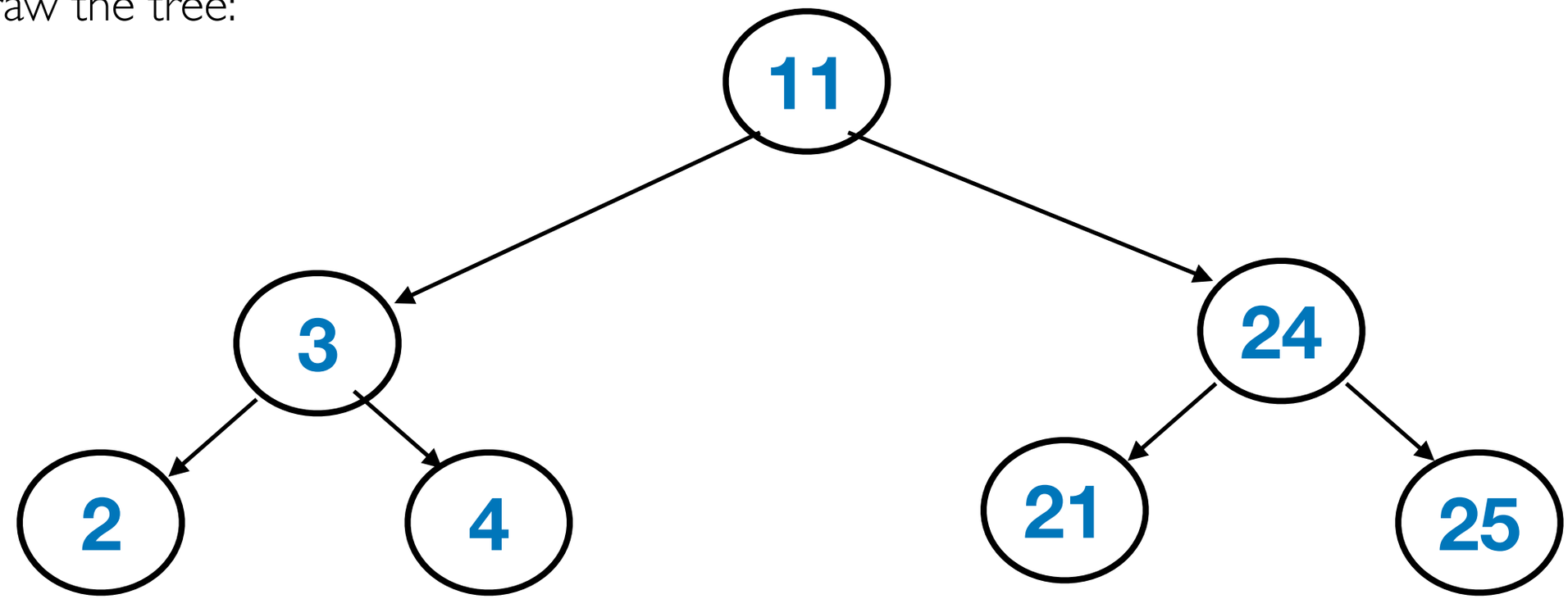
Draw the tree:



Review

Write down an insertion order that will produce a balanced tree...

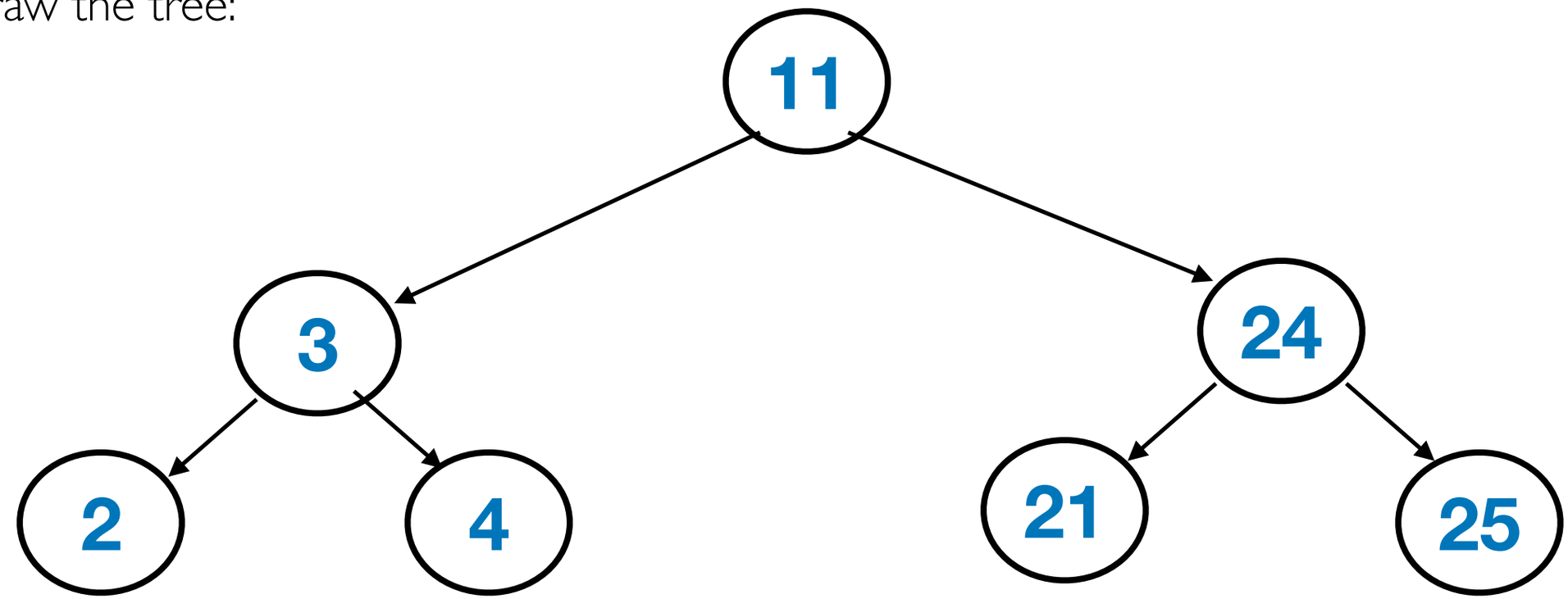
Draw the tree:



Review

Write down an insertion order that will produce a balanced tree...

Draw the tree:



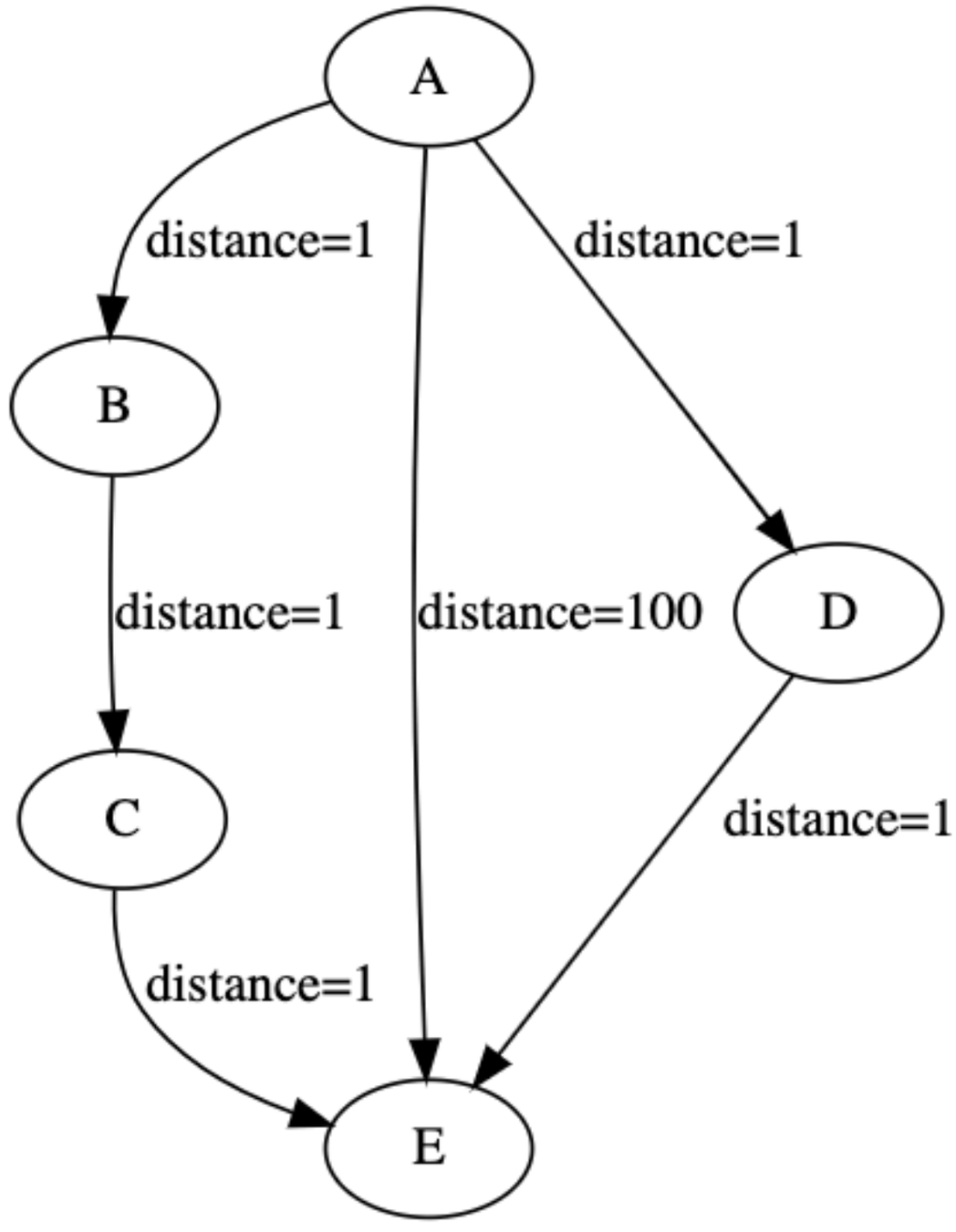
11, 3, 24, 2, 4, 21, 25

Shortest Weighted Path

What path will DFS choose?

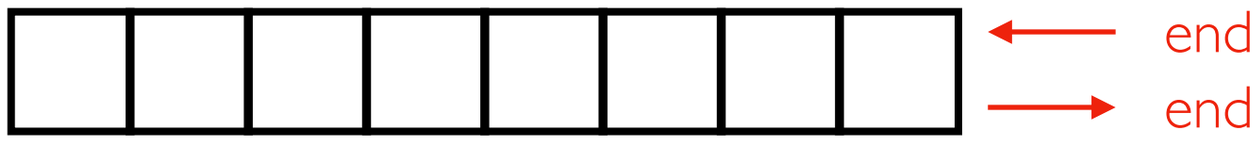
What path will BFS choose?

What path would you choose?

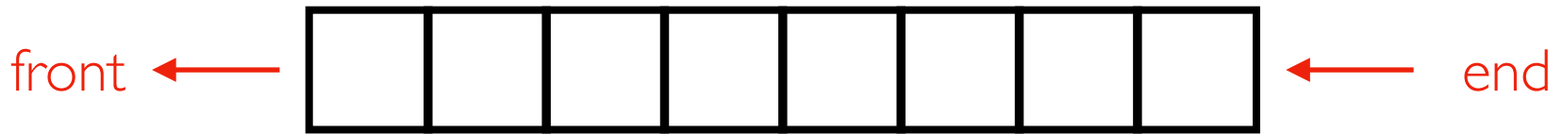


Your "to do" list: Stacks, Queues, and Priority Queues

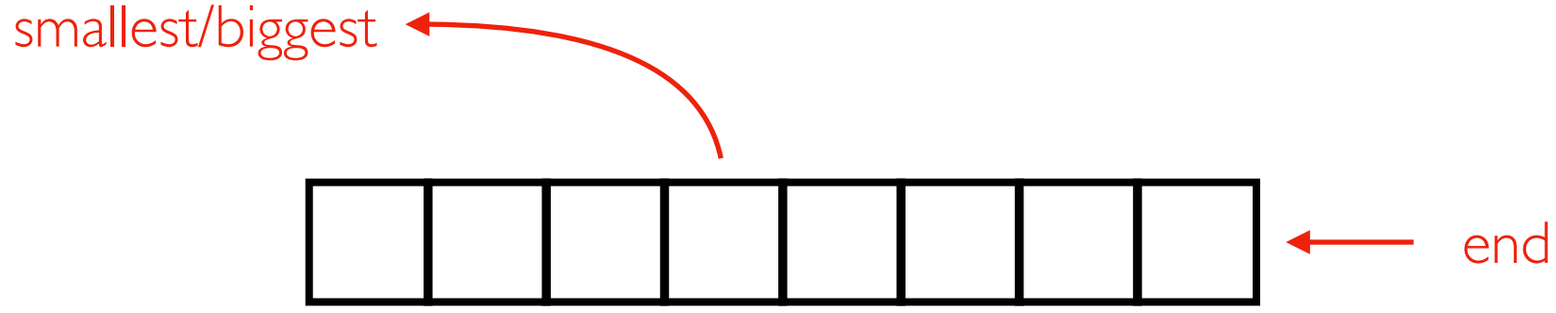
Stack



Queue



Priority Queue



Complexity: Time vs. Memory

```
def ratio_search(L, target):
    for n in L:
        for d in L:
            if n/d == target:
                return True
    return False
```

```
def list_ratios(L):
    ratios = []
    for n in L:
        for d in L:
            ratios.append(n/d)
    return ratios
```

if N is len(L) and f(N) is the **number of steps**, with is the Big-O complexity of each function?

Complexity: Time vs. Memory

```
def ratio_search(L, target):
    for n in L:
        for d in L:
            if n/d == target:
                return True
    return False
```

```
def list_ratios(L):
    ratios = []
    for n in L:
        for d in L:
            ratios.append(n/d)
    return ratios
```

if N is len(L) and f(N) is the **max memory** used, with is the Big-O complexity of each function?