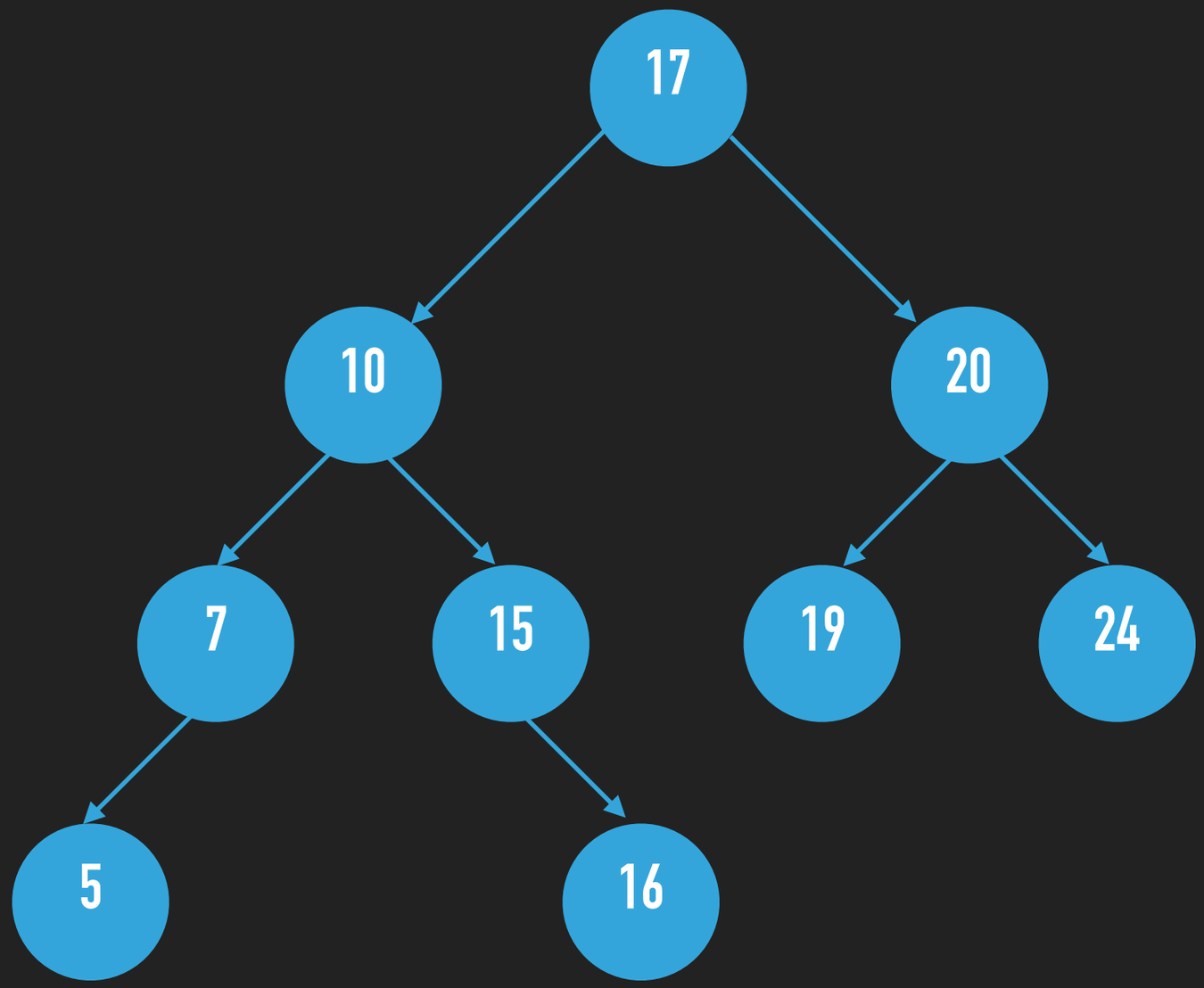
DG WEEK 5

RANK AND SELECT

Abdul Fatir Ansari

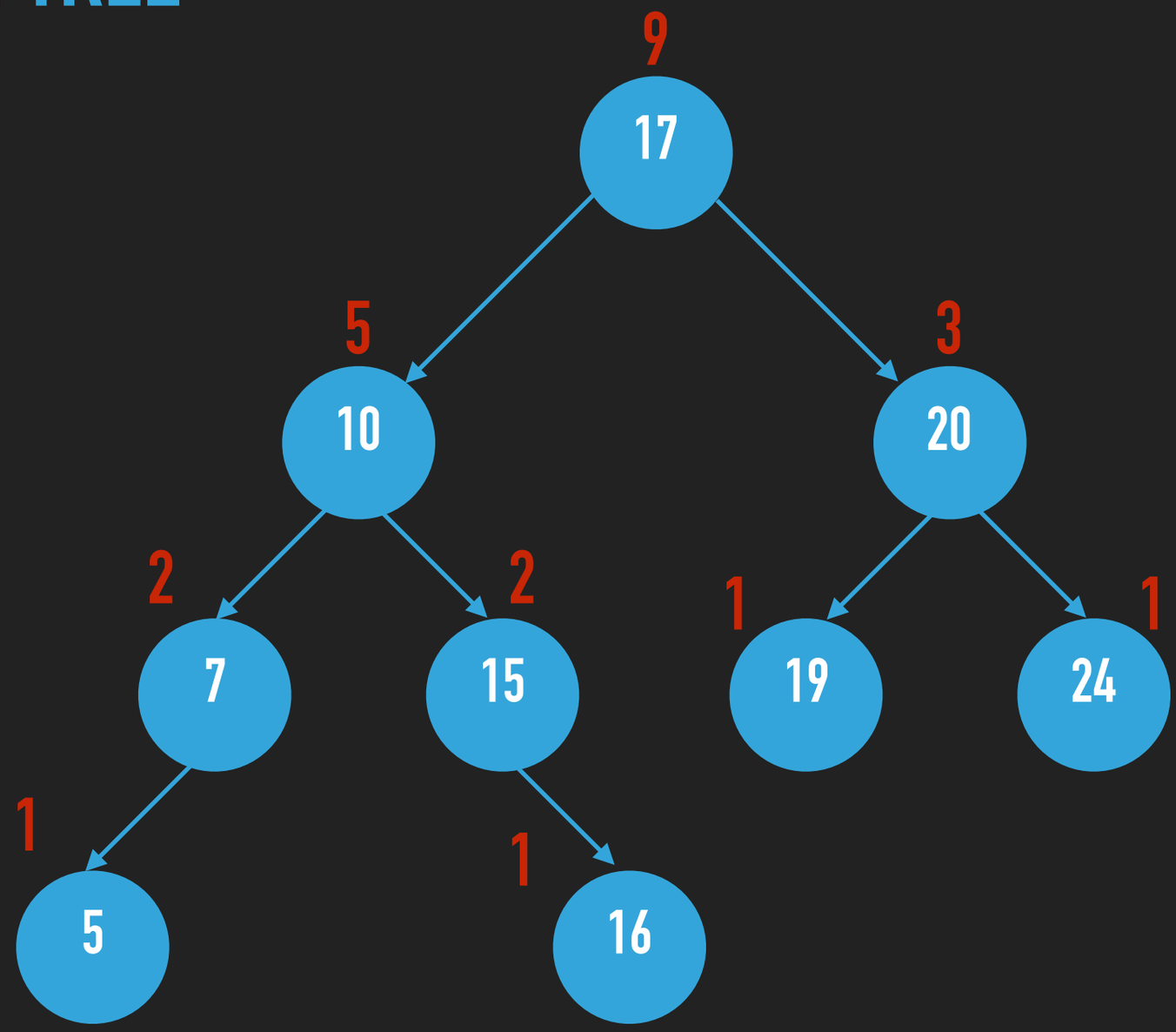
AUGMENTED BINARY SEARCH TREE

- Every node stores the size of subtree rooted at that node.



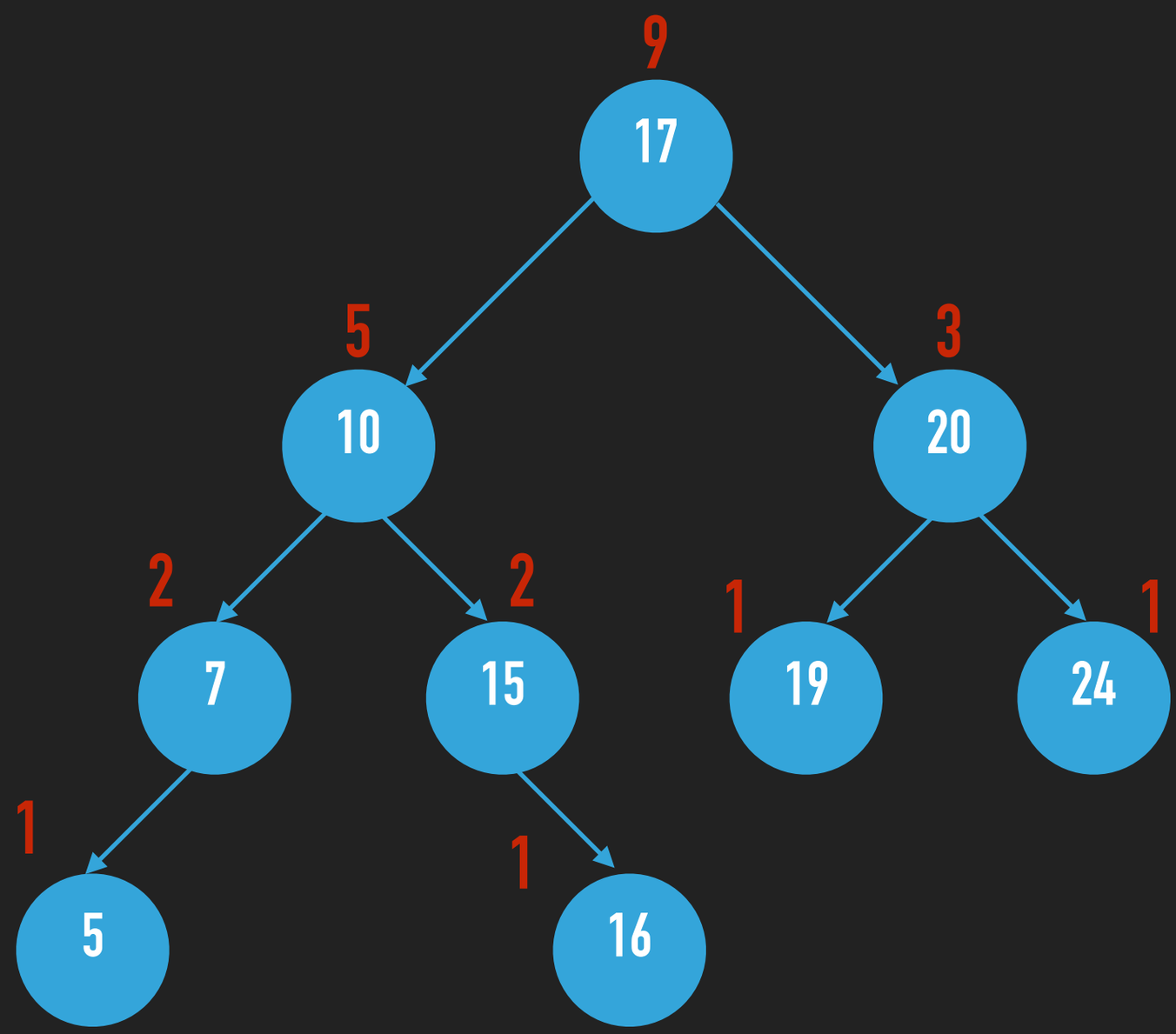
AUGMENTED BINARY SEARCH TREE

- Every node stores the size of subtree rooted at that node.
- Size(n) = Size(n.left) + Size(n.right) + 1



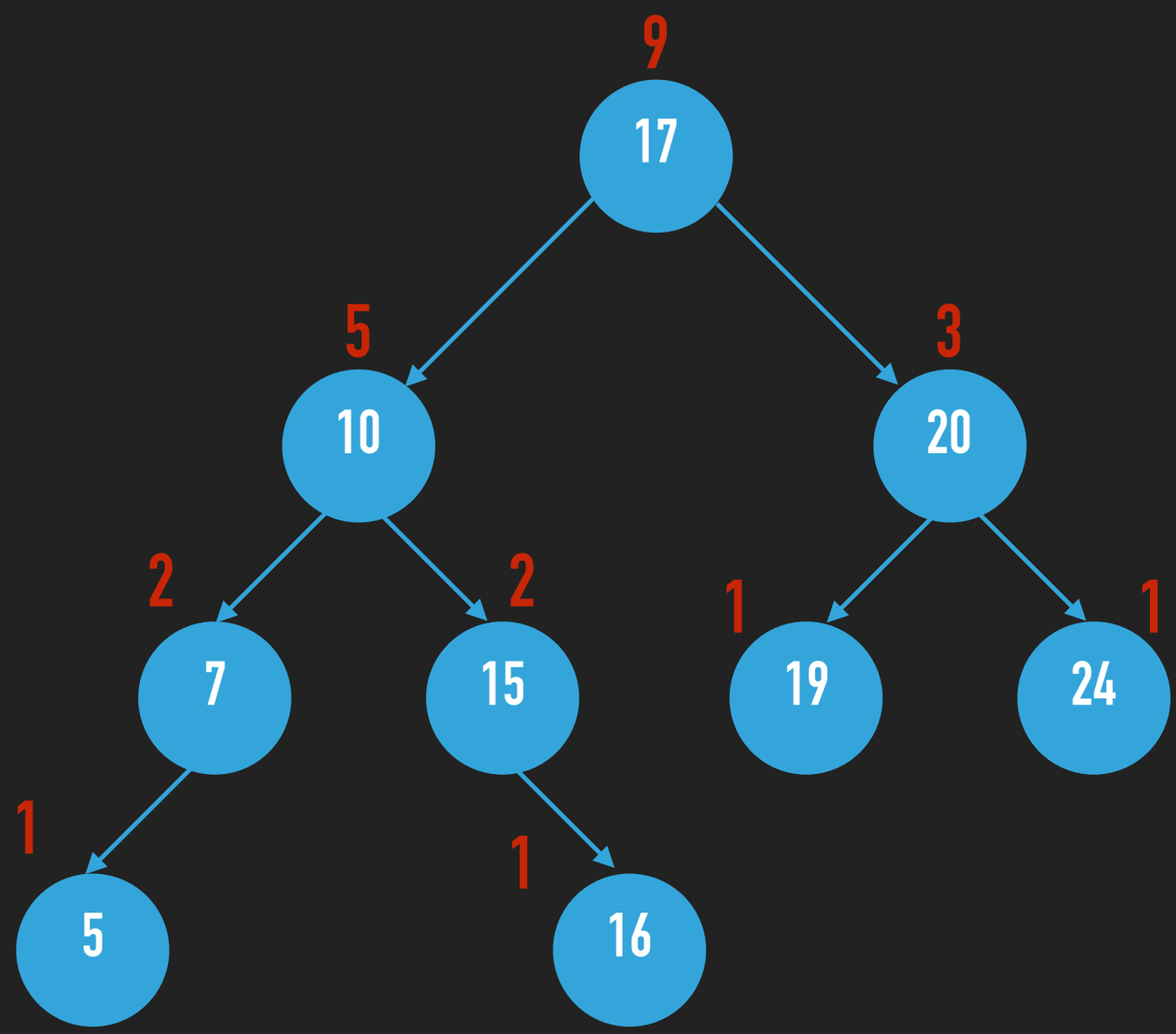
SELECT

- Select(5th)



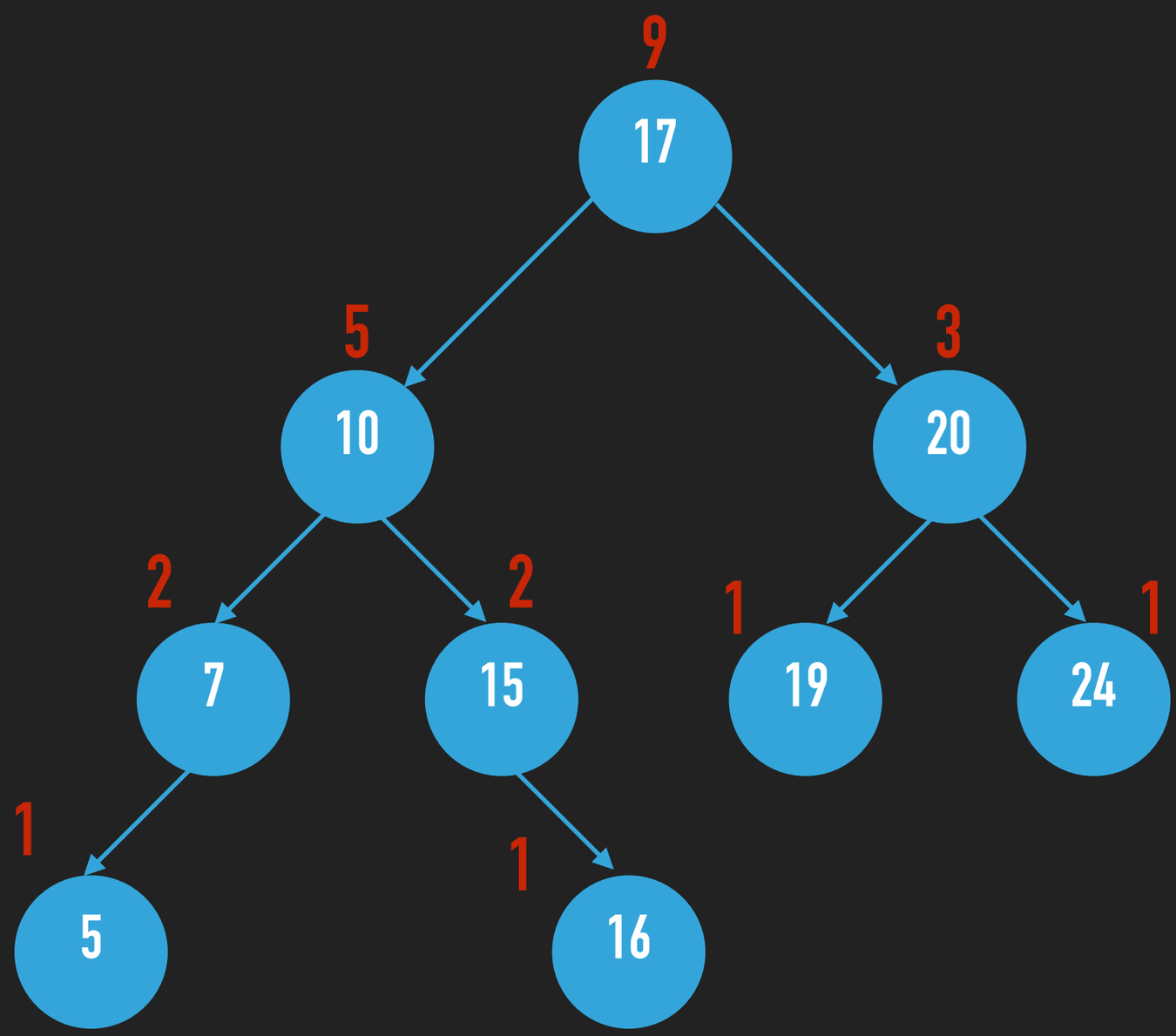
SELECT

- Select(8th)



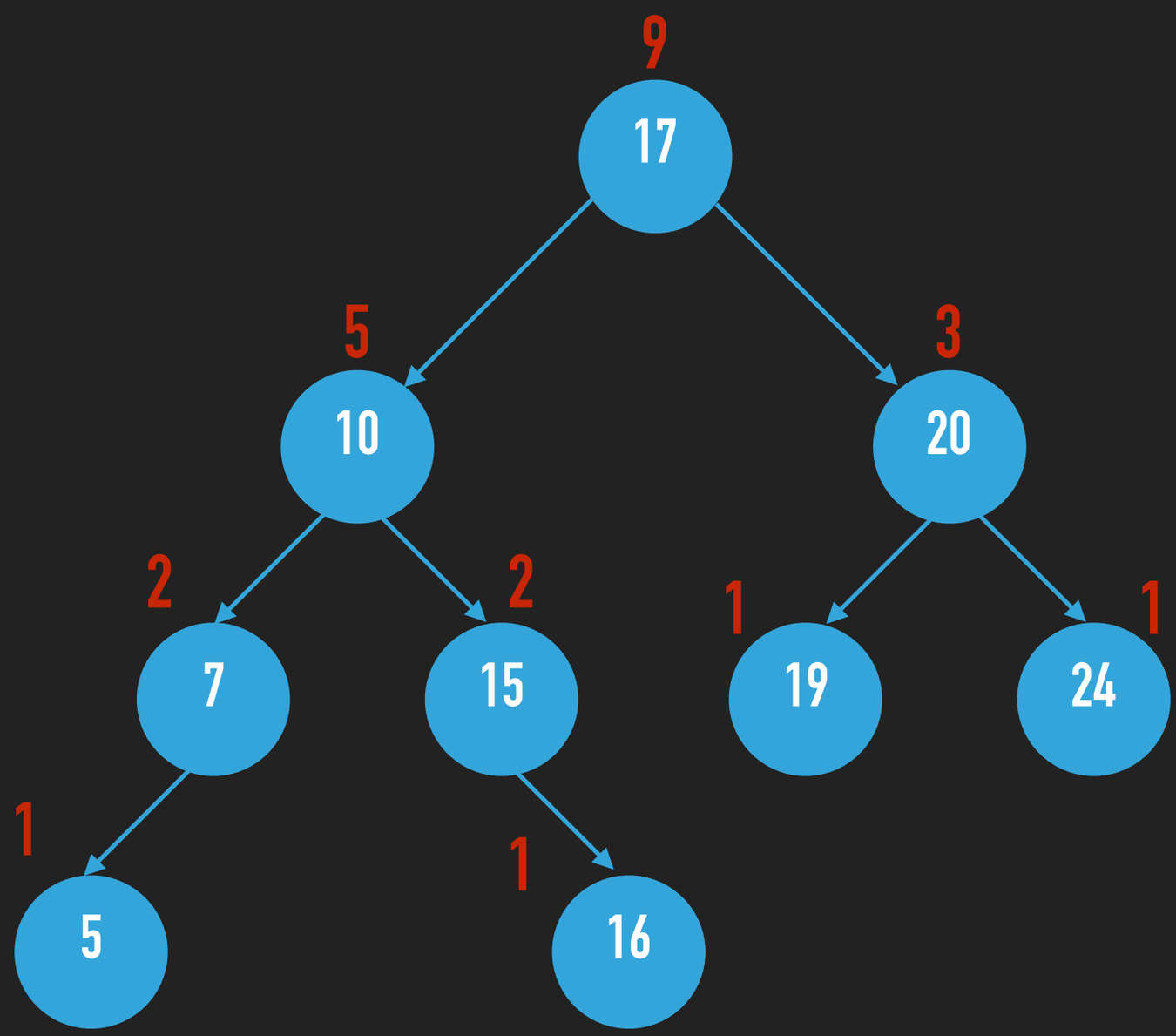
RANK

- Rank(10)



RANK

- Rank(20)



SELECT — PSEUDOCODE

- **Select**(T, n): # Selects n^{th} element in tree T
  - c = T.left.size + 1
  - **if** c == n: **return** T.val
  - **else if** n > c: **return Select**(T.right, n - c)
  - **else**: **return Select**(T.left, n)

RANK — PSEUDOCODE

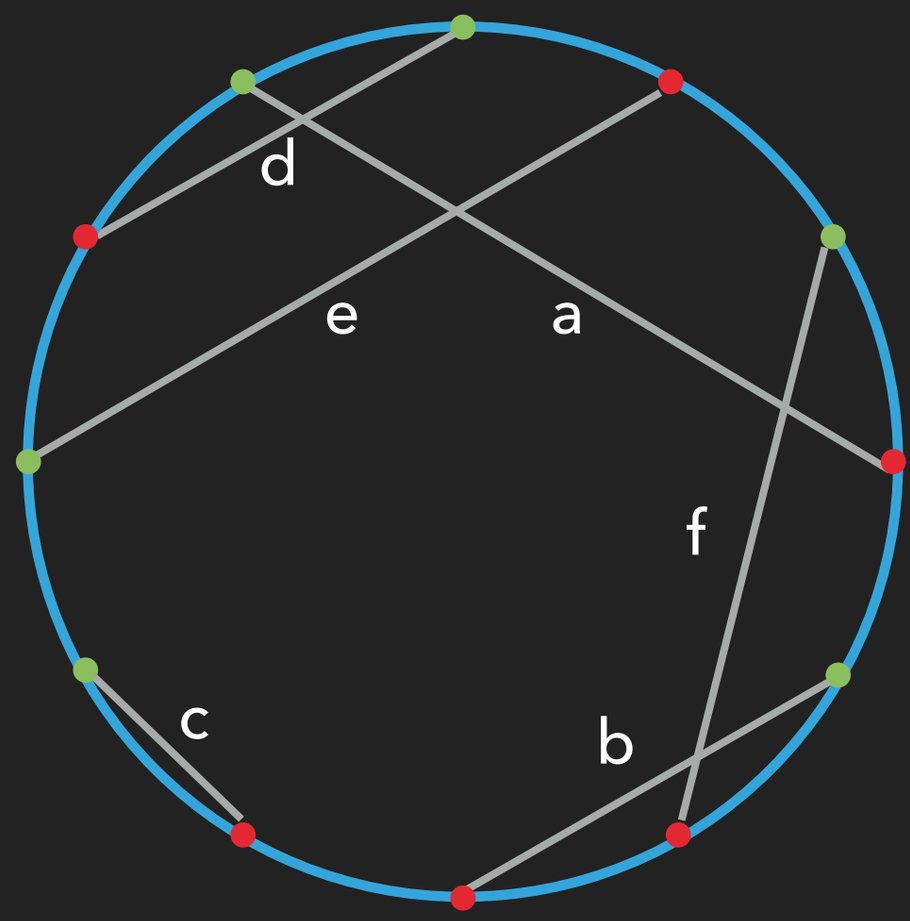
- **Rank**(T, a): # Finds the rank of a in tree T
  - rank = a.left.size + 1
  - b = a
  - **while** b *is not* the T.root:
    - **if** b *is* b.parent.right:
      - rank += b.parent.left.size + 1
    - b = b.parent
  - **return** rank

CIRCLE INTERSECTIONS

CIRCLE INTERSECTIONS

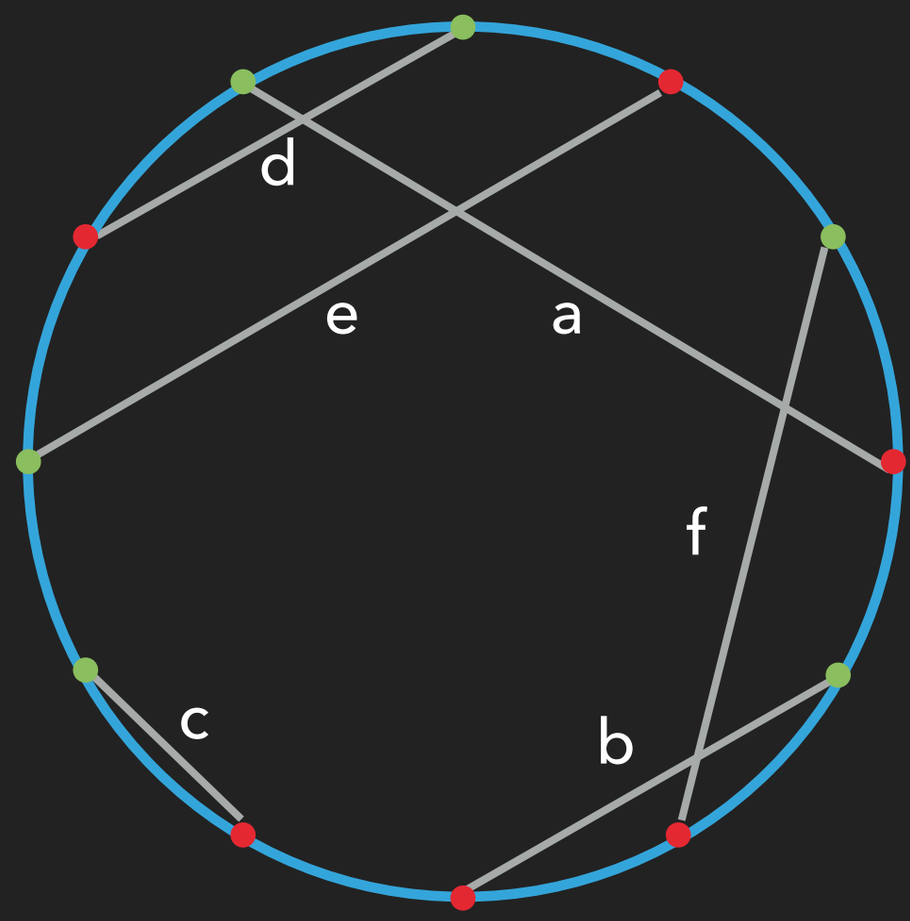
Find the number of intersections

in $O(n \log n)$ time.

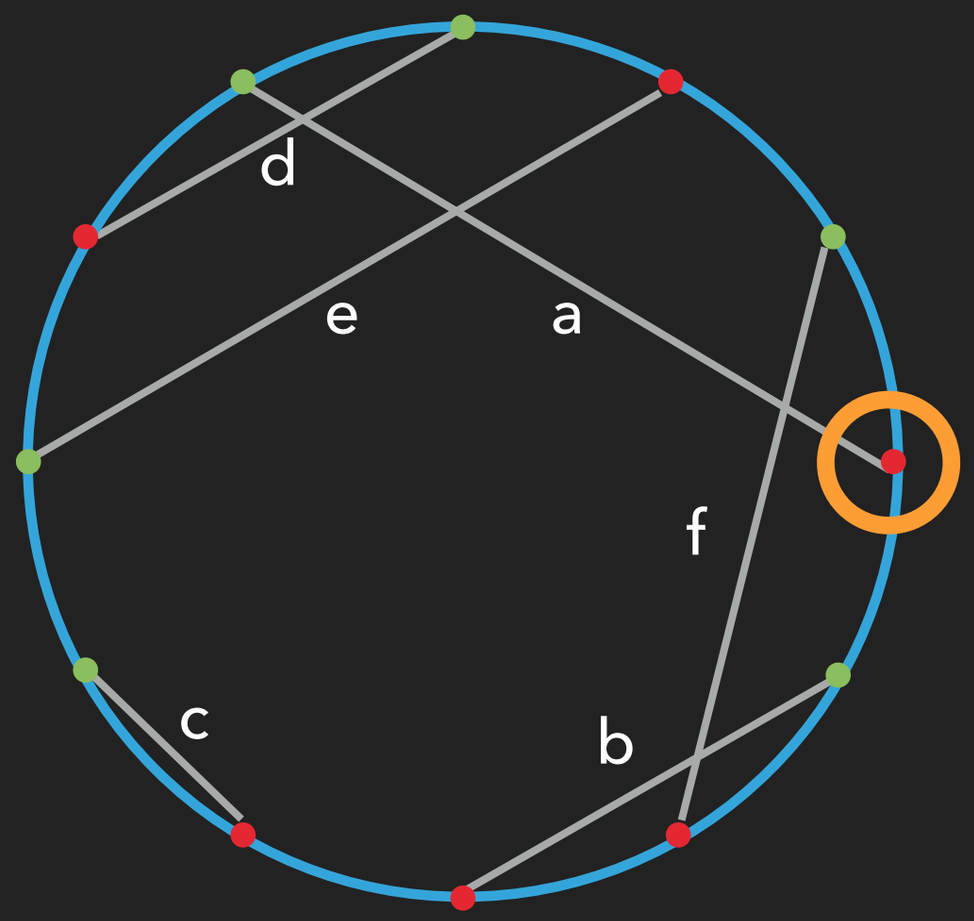


CIRCLE INTERSECTIONS

We'll keep a list sorted by angle.

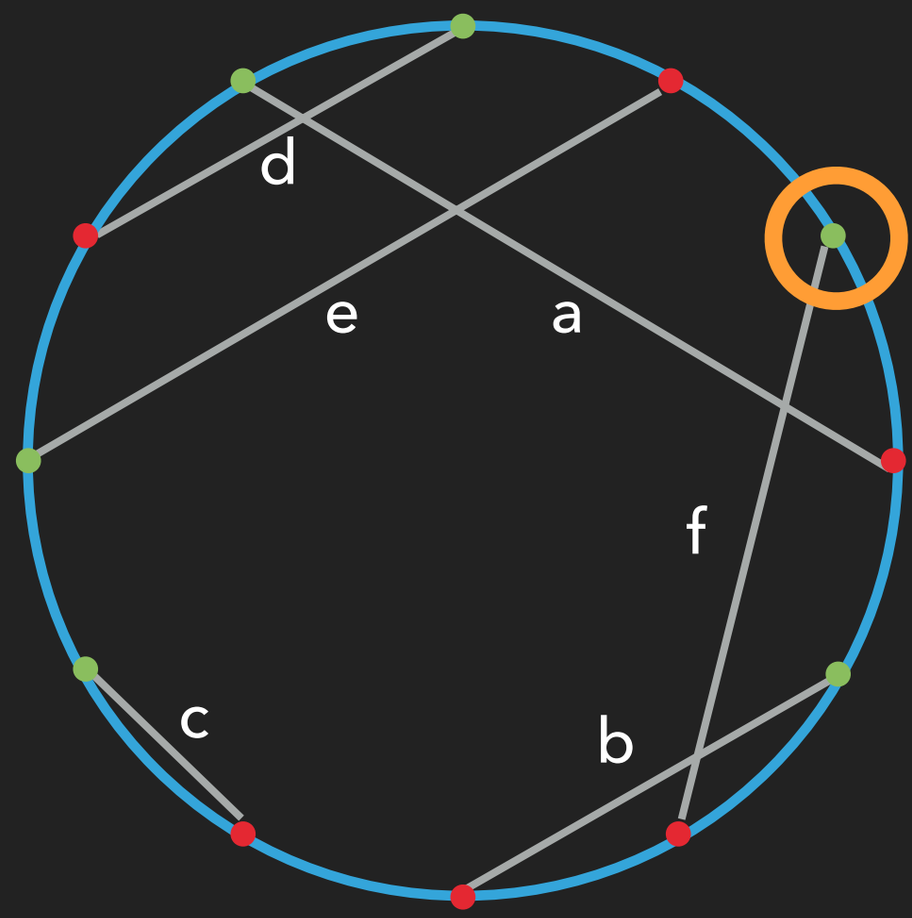


CIRCLE INTERSECTIONS



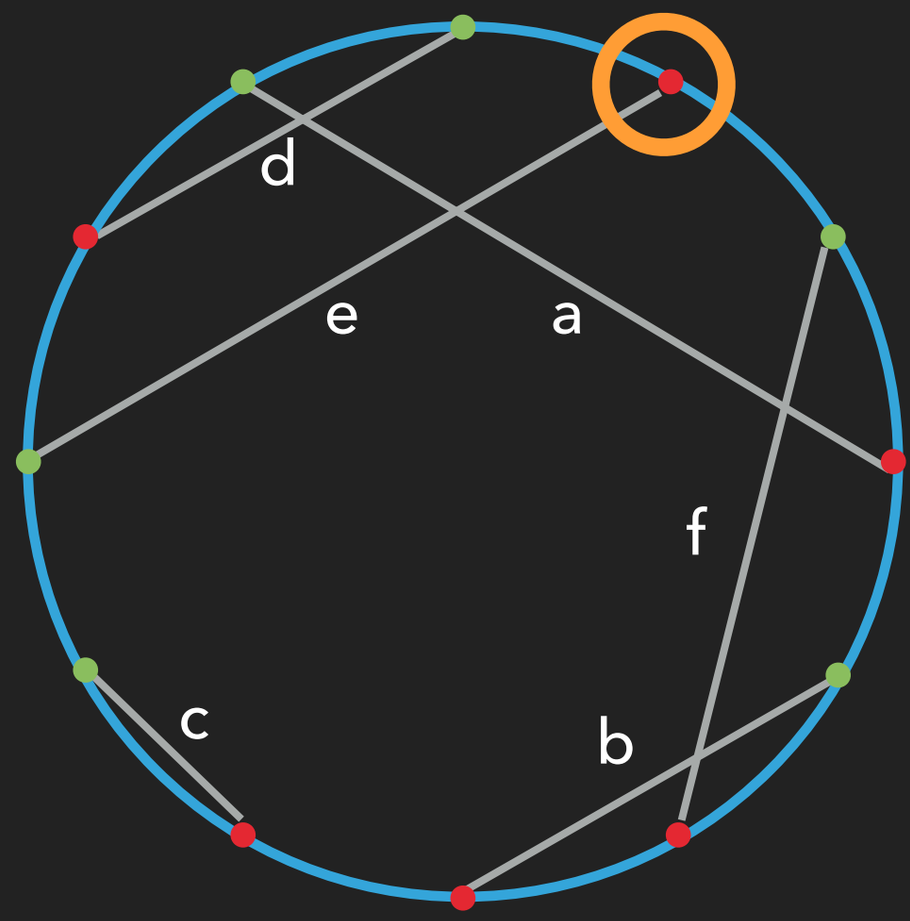
TOTAL INTERSECTIONS — 0

CIRCLE INTERSECTIONS



TOTAL INTERSECTIONS — 0

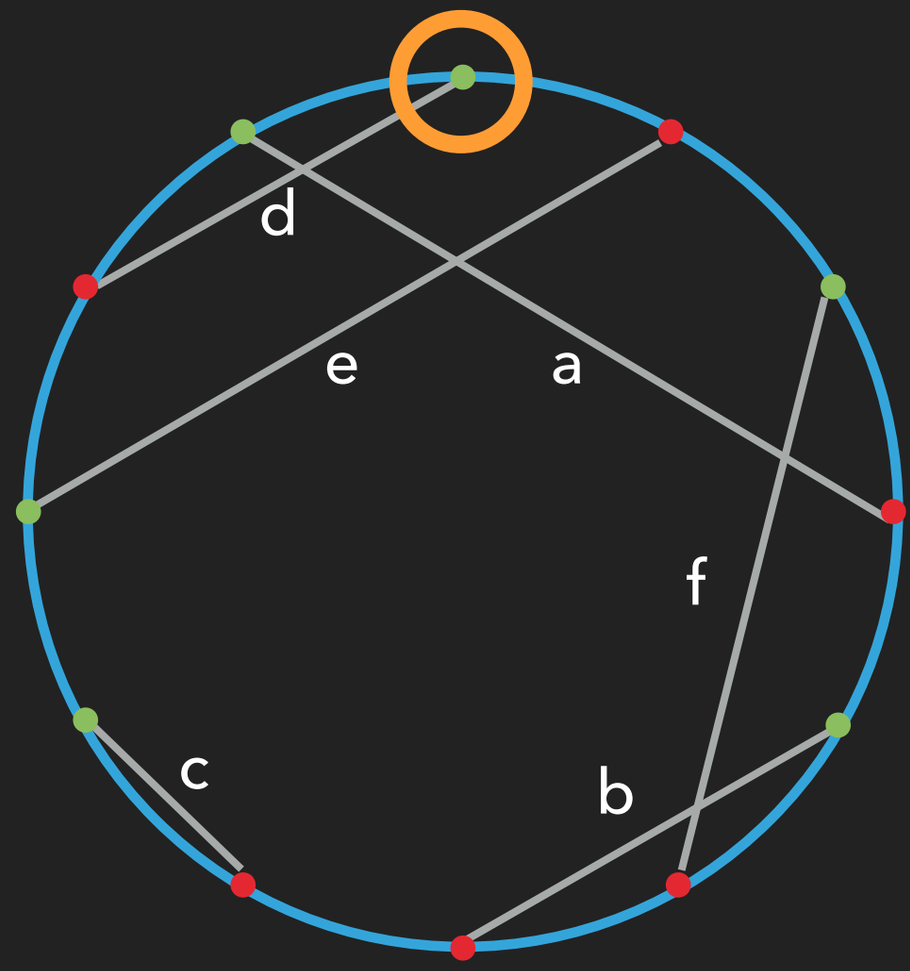
CIRCLE INTERSECTIONS



TOTAL INTERSECTIONS — 0

CIRCLE INTERSECTIONS

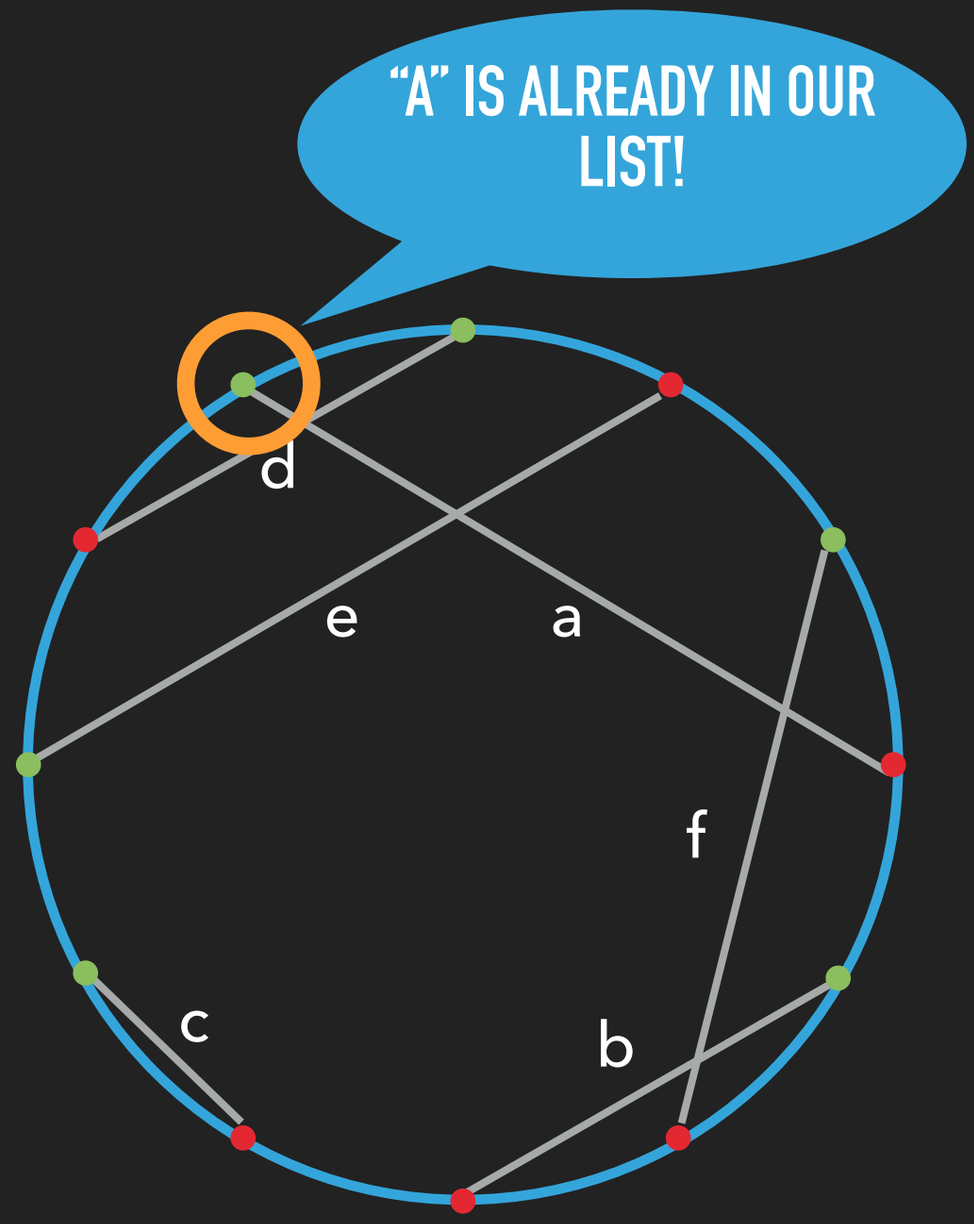
A	F	E			



TOTAL INTERSECTIONS — 0

CIRCLE INTERSECTIONS

A	F	E	D		

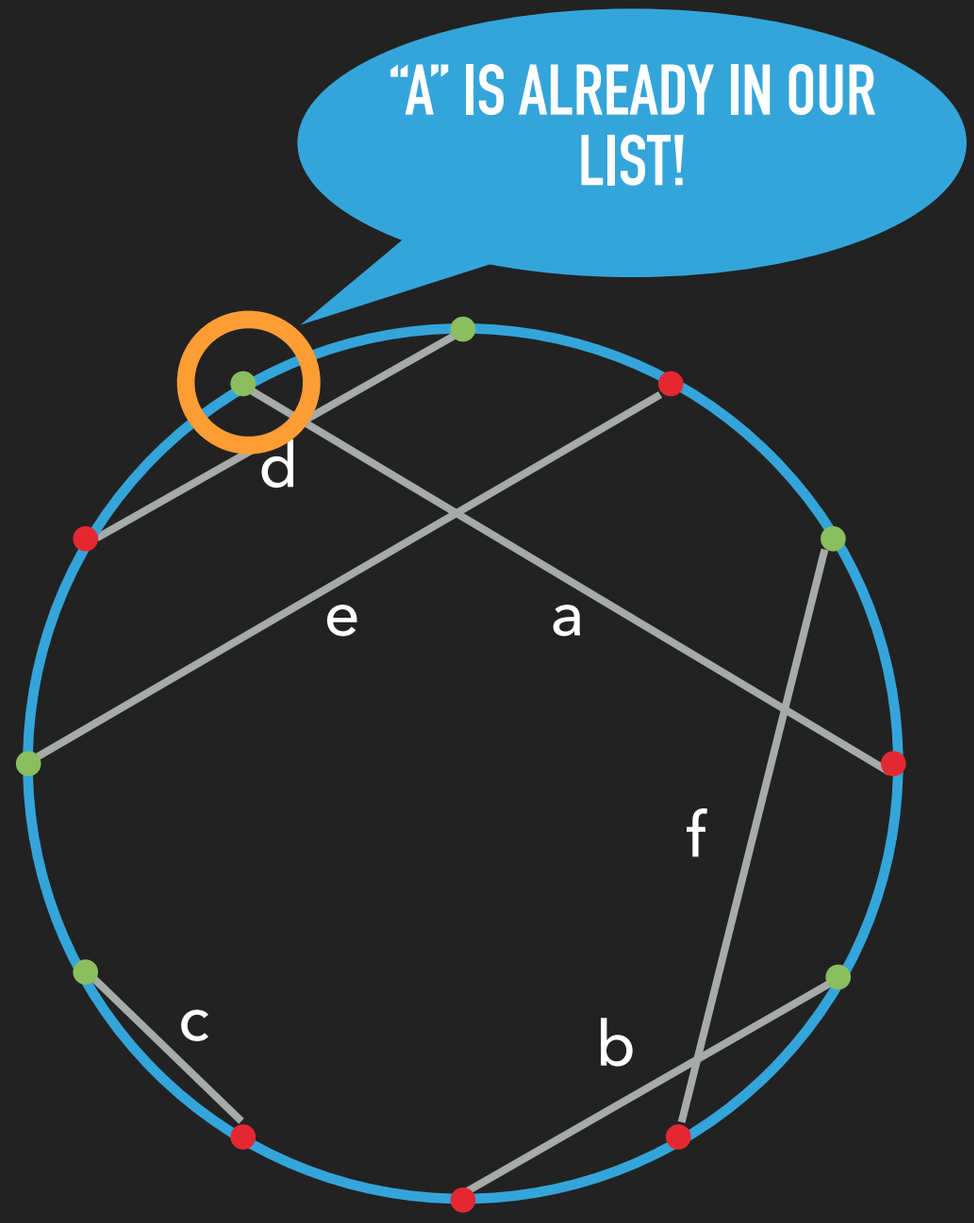


TOTAL INTERSECTIONS — 0

CIRCLE INTERSECTIONS

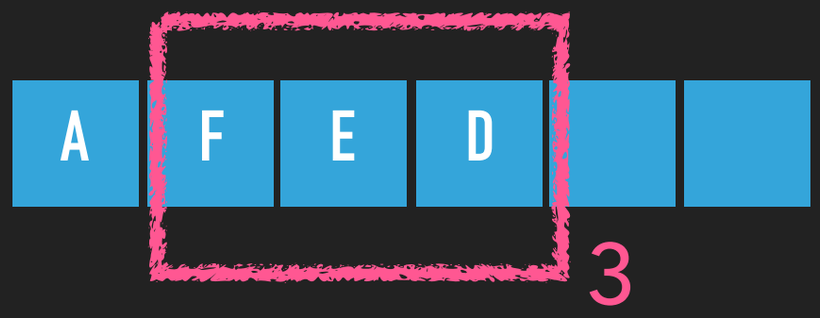
A	F	E	D		

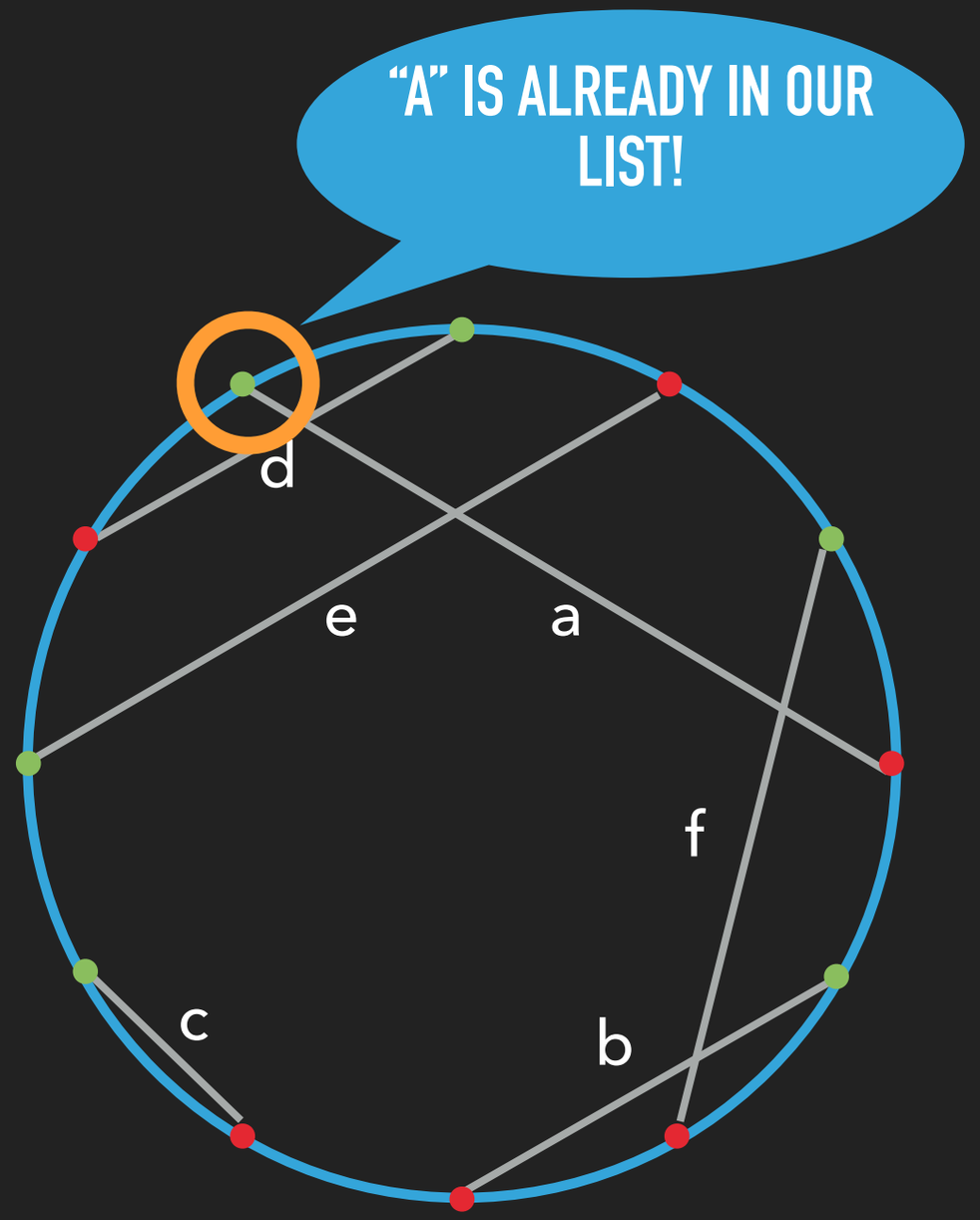
HOW MANY LINE SEGMENTS BEGAN AFTER “A” THAT HAVEN’T YET ENDED?



TOTAL INTERSECTIONS — 0

CIRCLE INTERSECTIONS



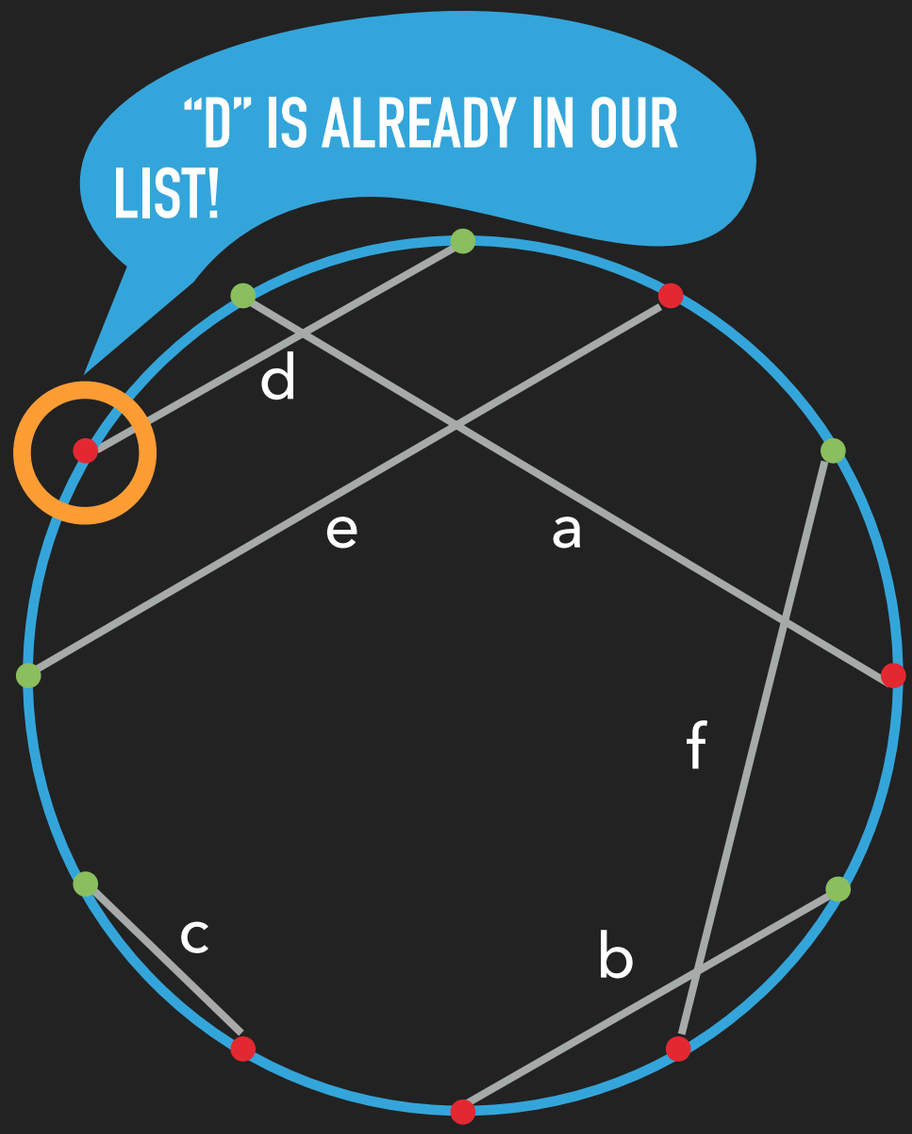


HOW MANY LINE SEGMENTS BEGAN AFTER “A” THAT HAVEN’T YET ENDED?

TOTAL INTERSECTIONS — 3

CIRCLE INTERSECTIONS

F	E	D		

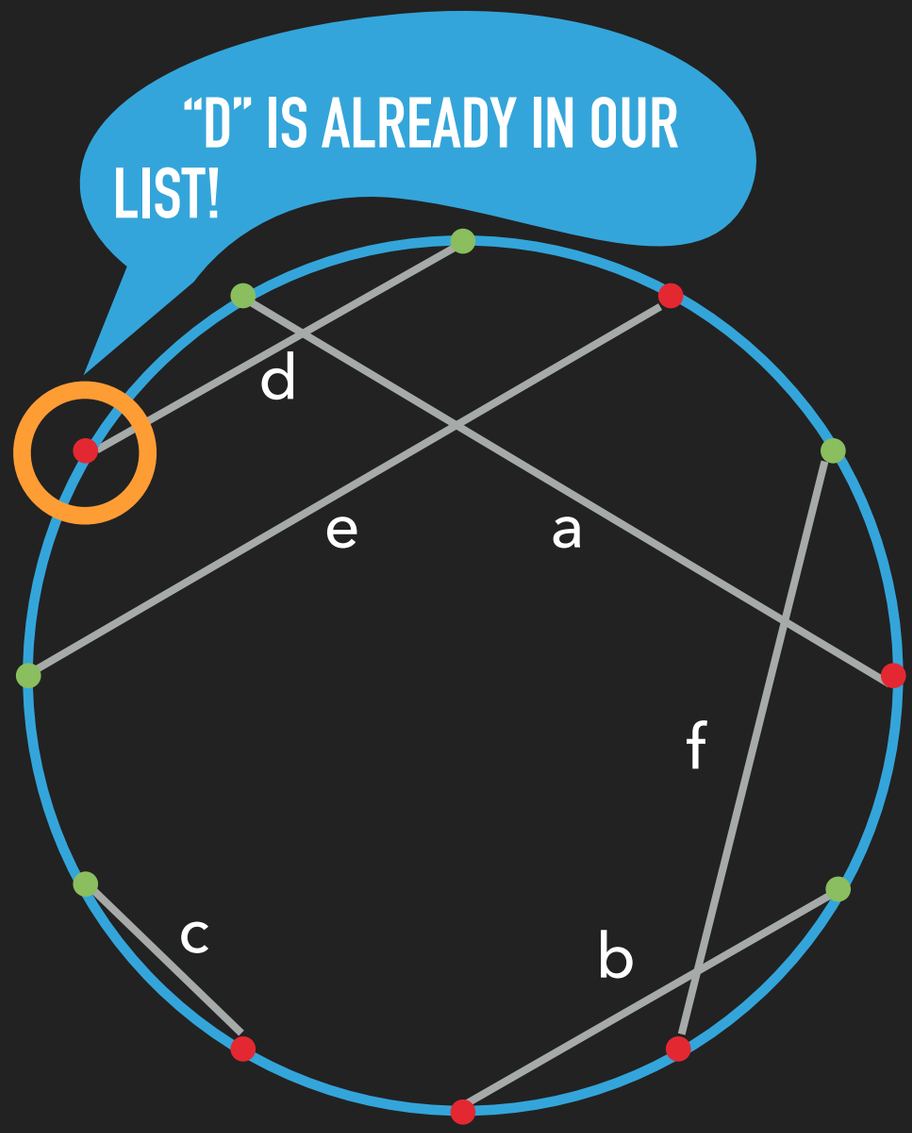


HOW MANY LINE SEGMENTS BEGAN AFTER “D” THAT HAVEN’T YET ENDED?

TOTAL INTERSECTIONS — 3

CIRCLE INTERSECTIONS

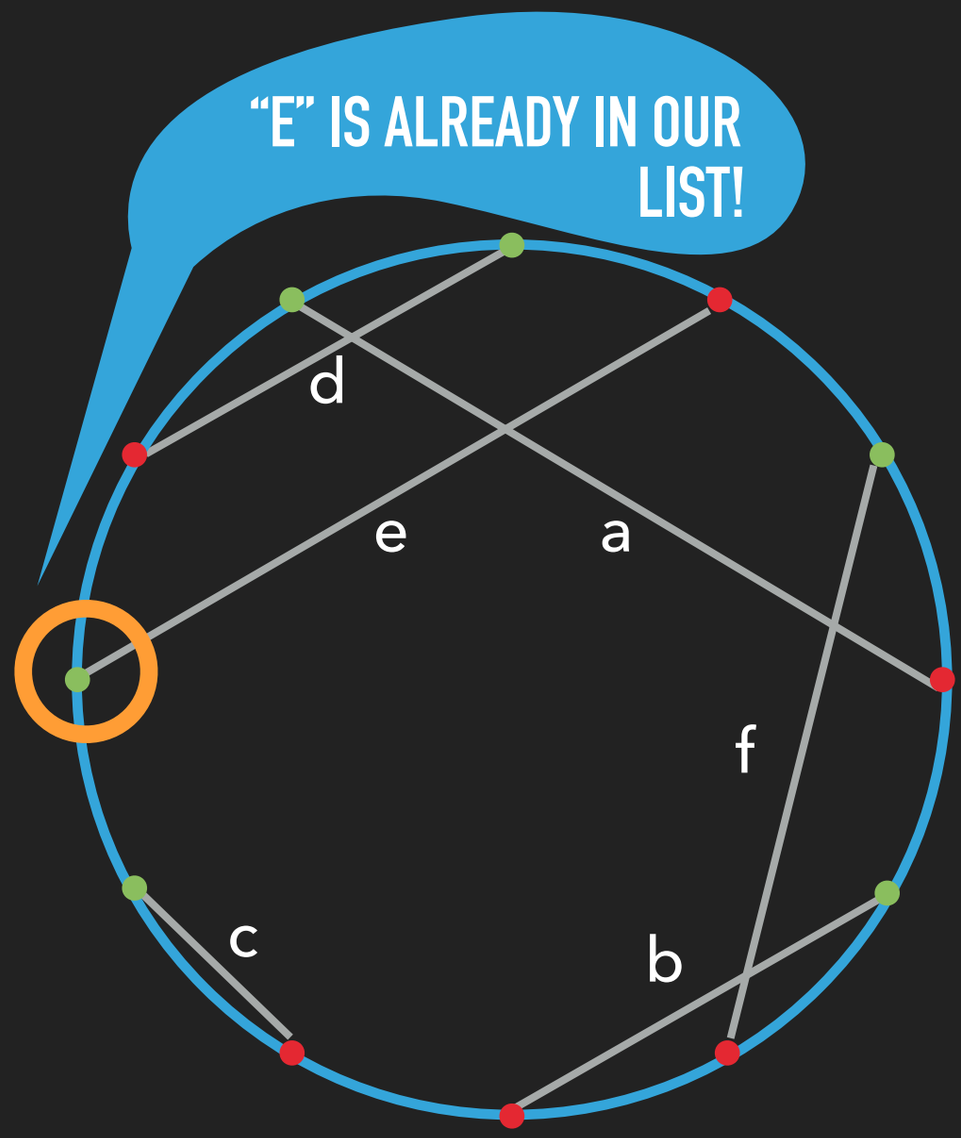
HOW MANY LINE SEGMENTS BEGAN AFTER “D” THAT HAVEN’T YET ENDED?



TOTAL INTERSECTIONS — 3

CIRCLE INTERSECTIONS

F	E			

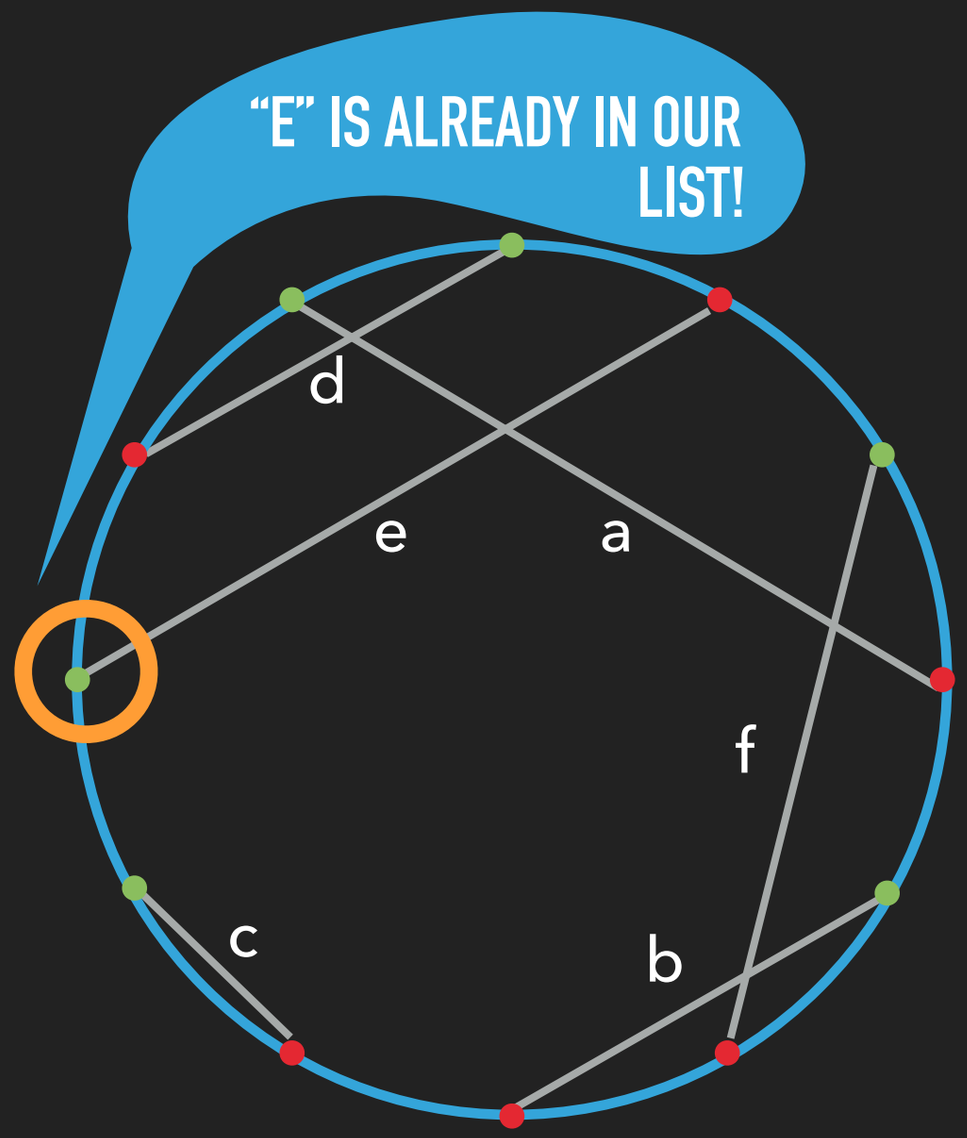


HOW MANY LINE SEGMENTS BEGAN AFTER “E” THAT HAVEN’T YET ENDED?

TOTAL INTERSECTIONS — 3

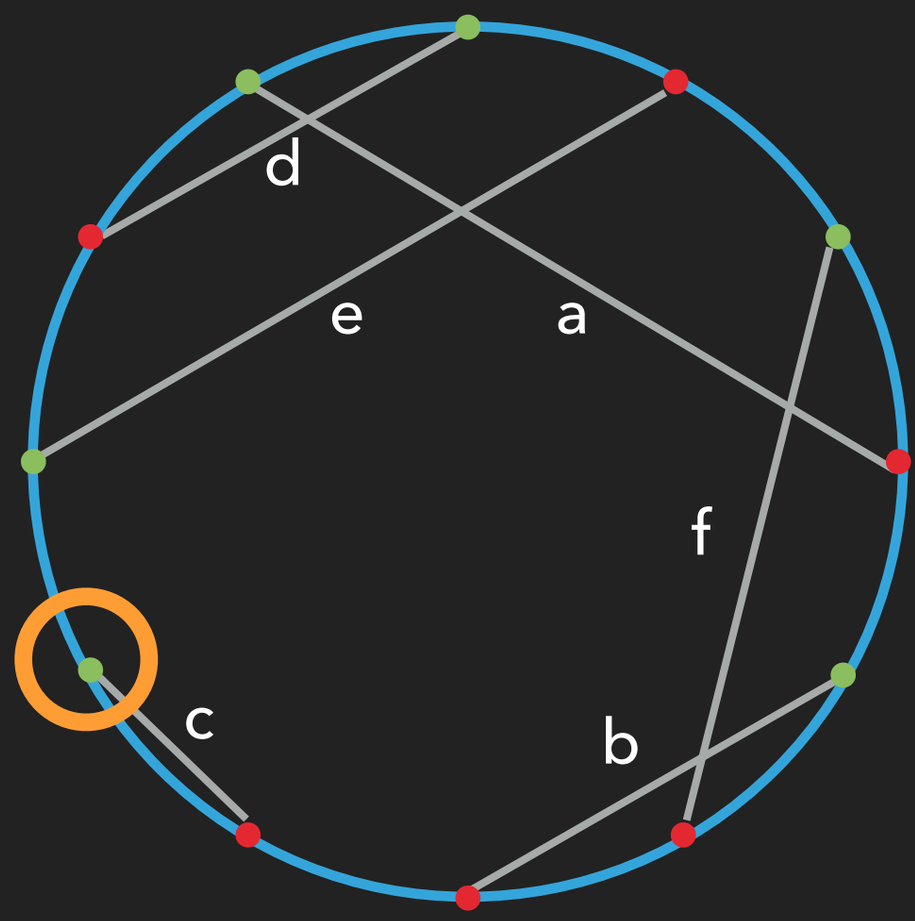
CIRCLE INTERSECTIONS

HOW MANY LINE SEGMENTS BEGAN AFTER “E” THAT HAVEN’T YET ENDED?



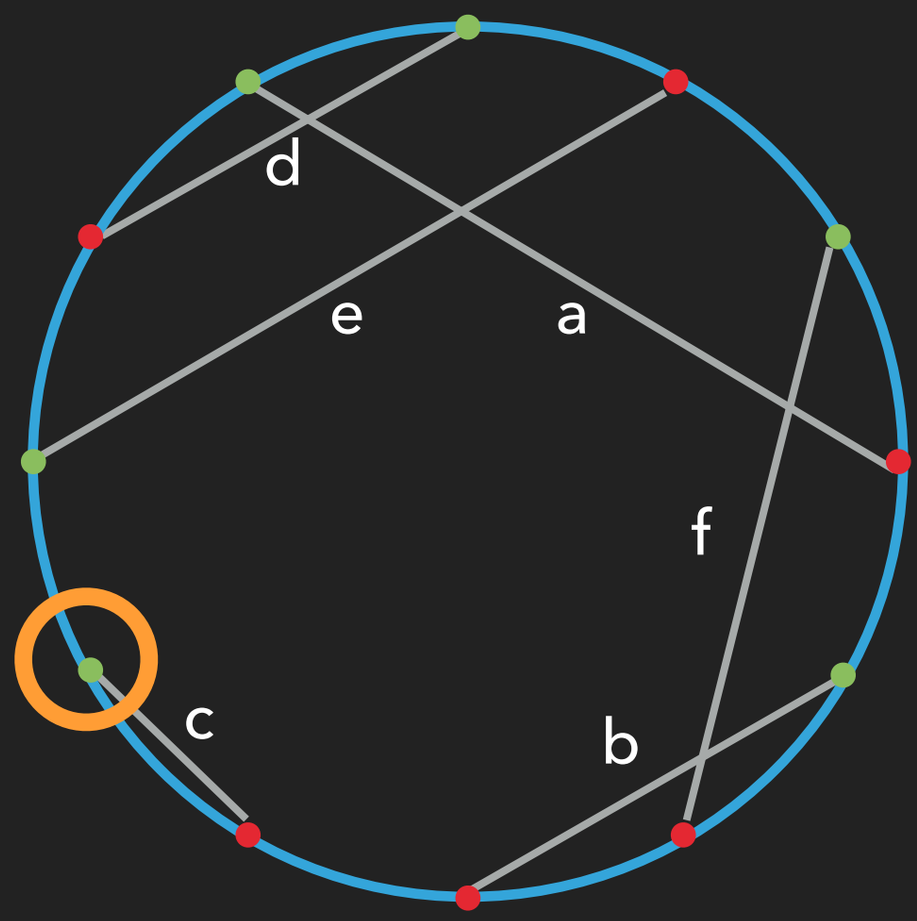
TOTAL INTERSECTIONS — 3

CIRCLE INTERSECTIONS



TOTAL INTERSECTIONS — 3

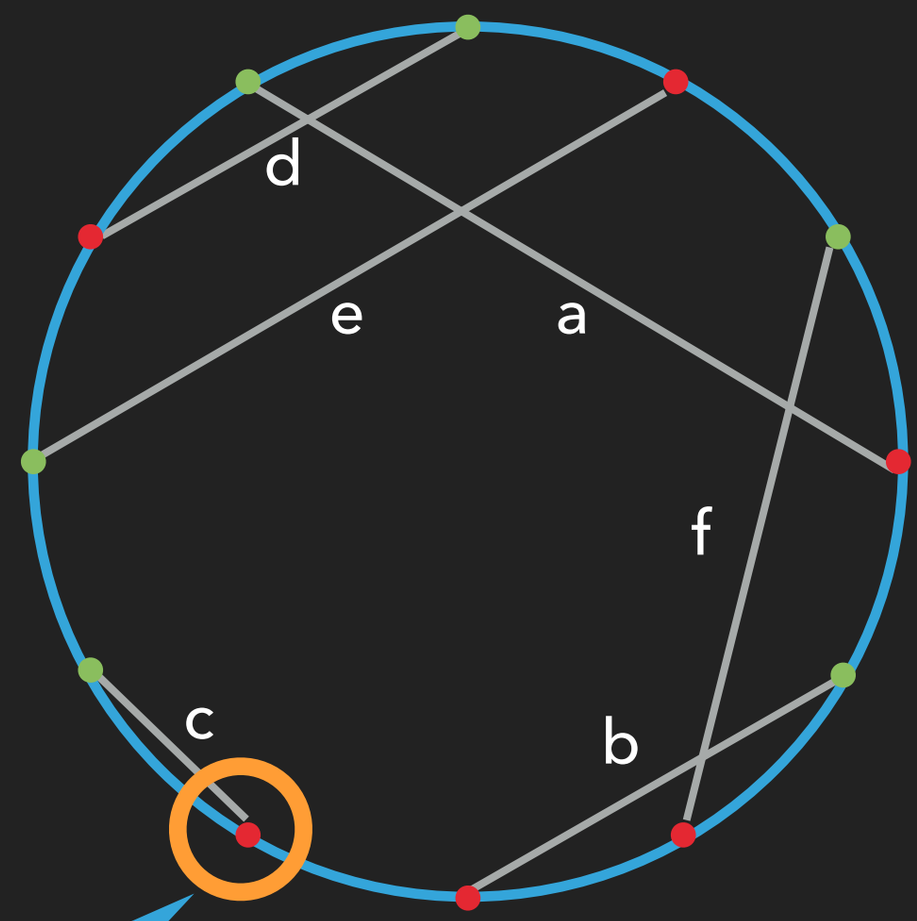
CIRCLE INTERSECTIONS



TOTAL INTERSECTIONS — 3

CIRCLE INTERSECTIONS

F	C			

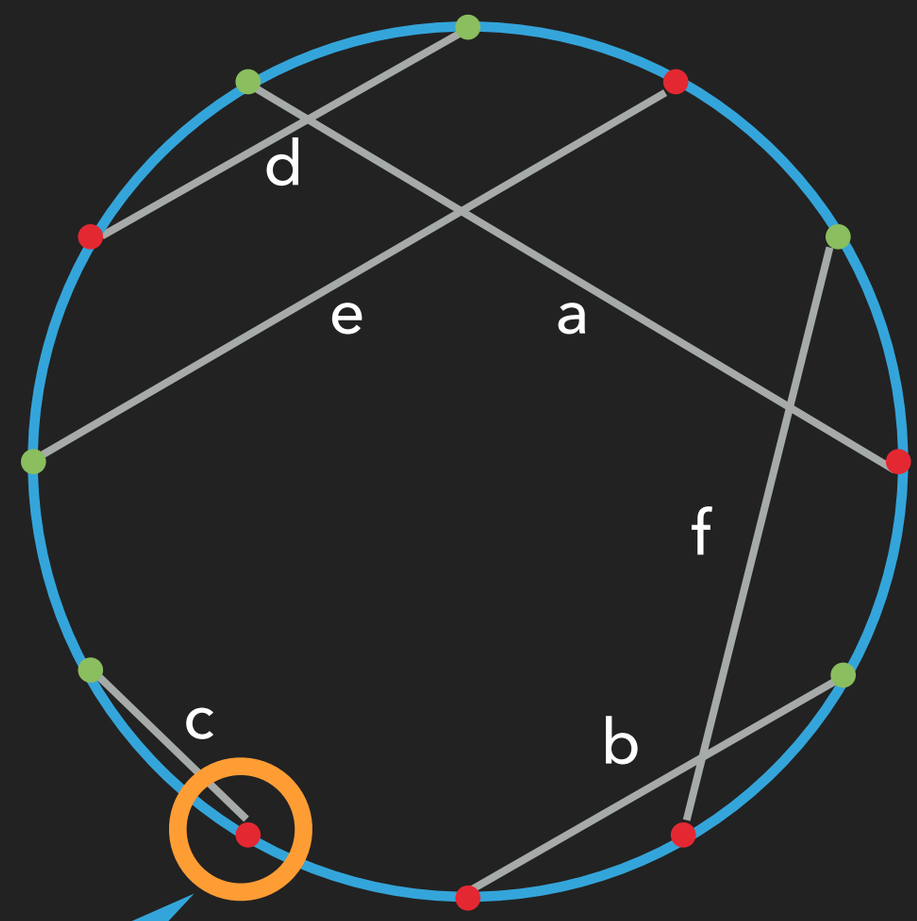


"C"
IS ALREADY IN OUR LIST!

TOTAL INTERSECTIONS — 3

CIRCLE INTERSECTIONS

F	C			



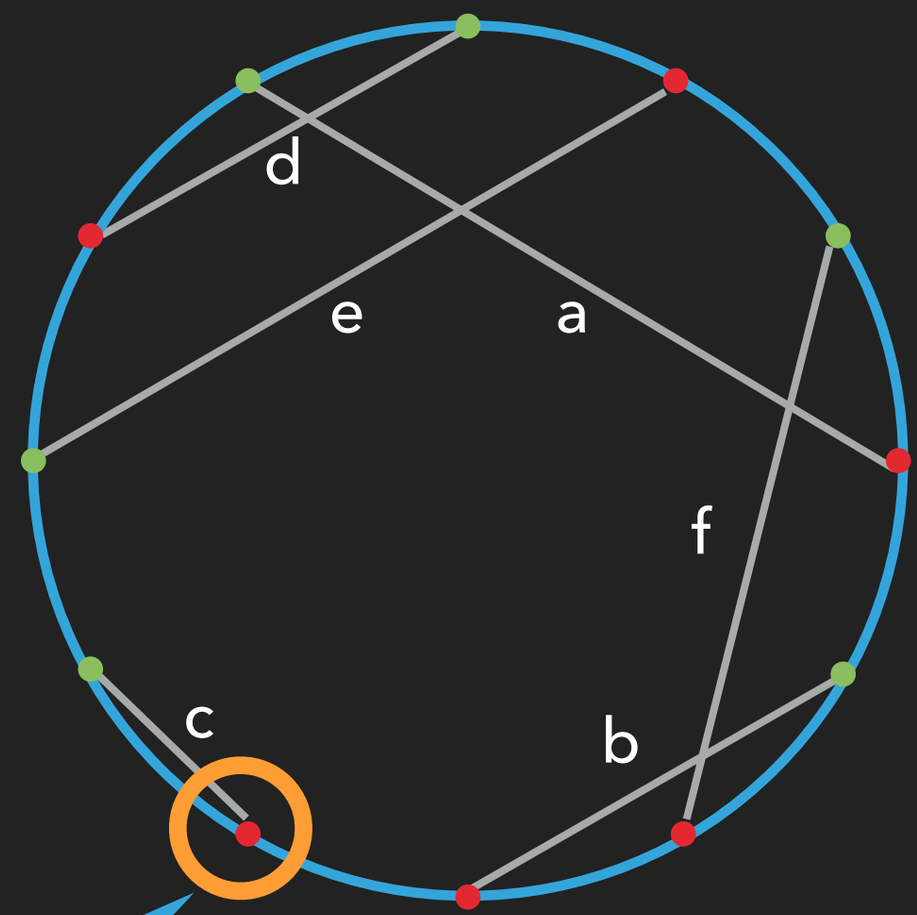
HOW MANY LINE SEGMENTS BEGAN AFTER “C” THAT HAVEN’T YET ENDED?

“C” IS ALREADY IN OUR LIST!

TOTAL INTERSECTIONS — 3

CIRCLE INTERSECTIONS

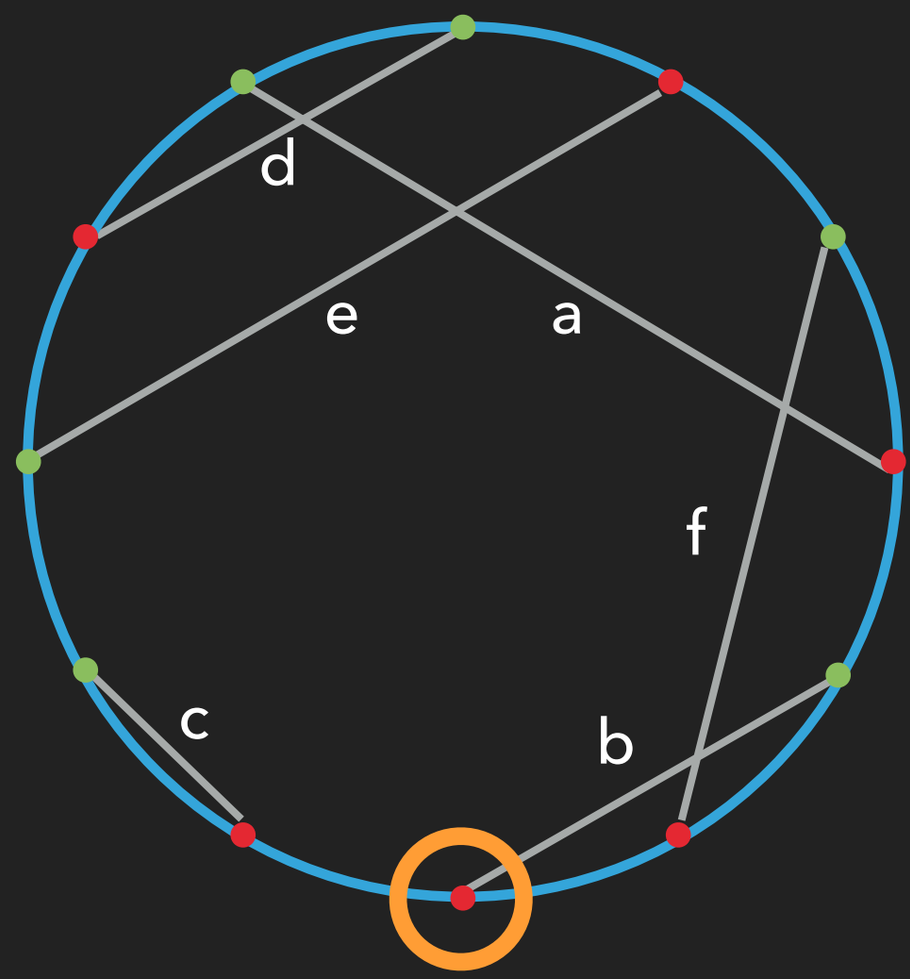
HOW MANY LINE SEGMENTS BEGAN AFTER “C” THAT HAVEN’T YET ENDED?



“C” IS ALREADY IN OUR LIST!

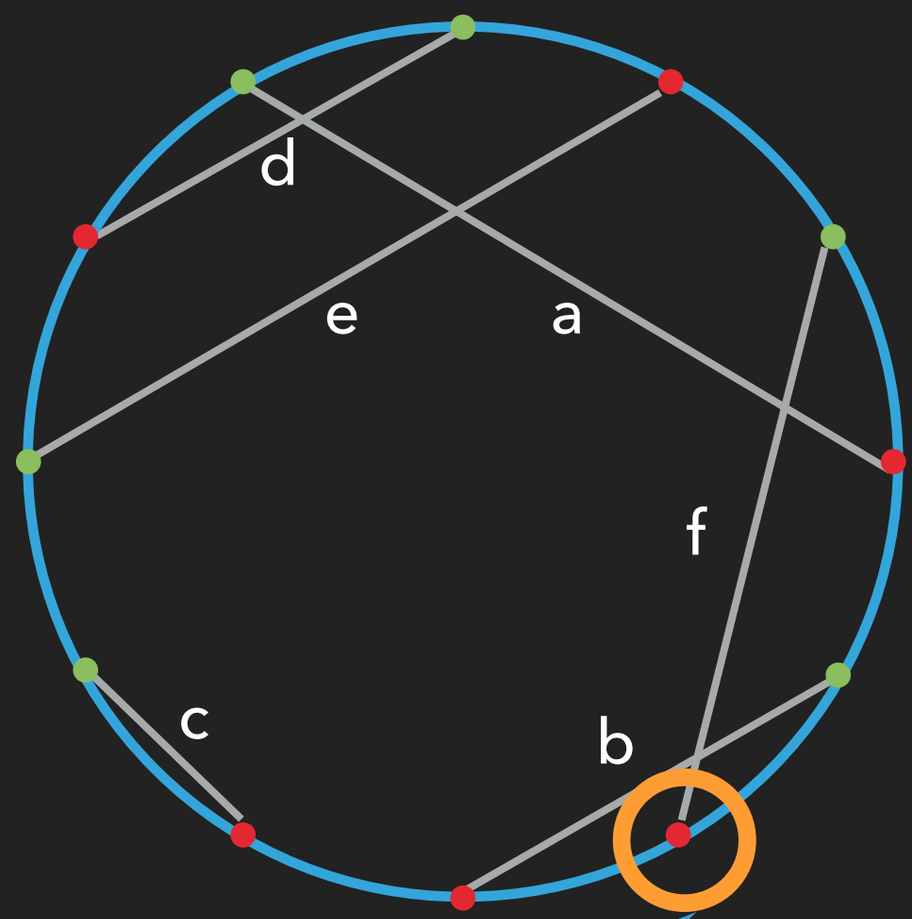
TOTAL INTERSECTIONS — 3

CIRCLE INTERSECTIONS



TOTAL INTERSECTIONS — 3

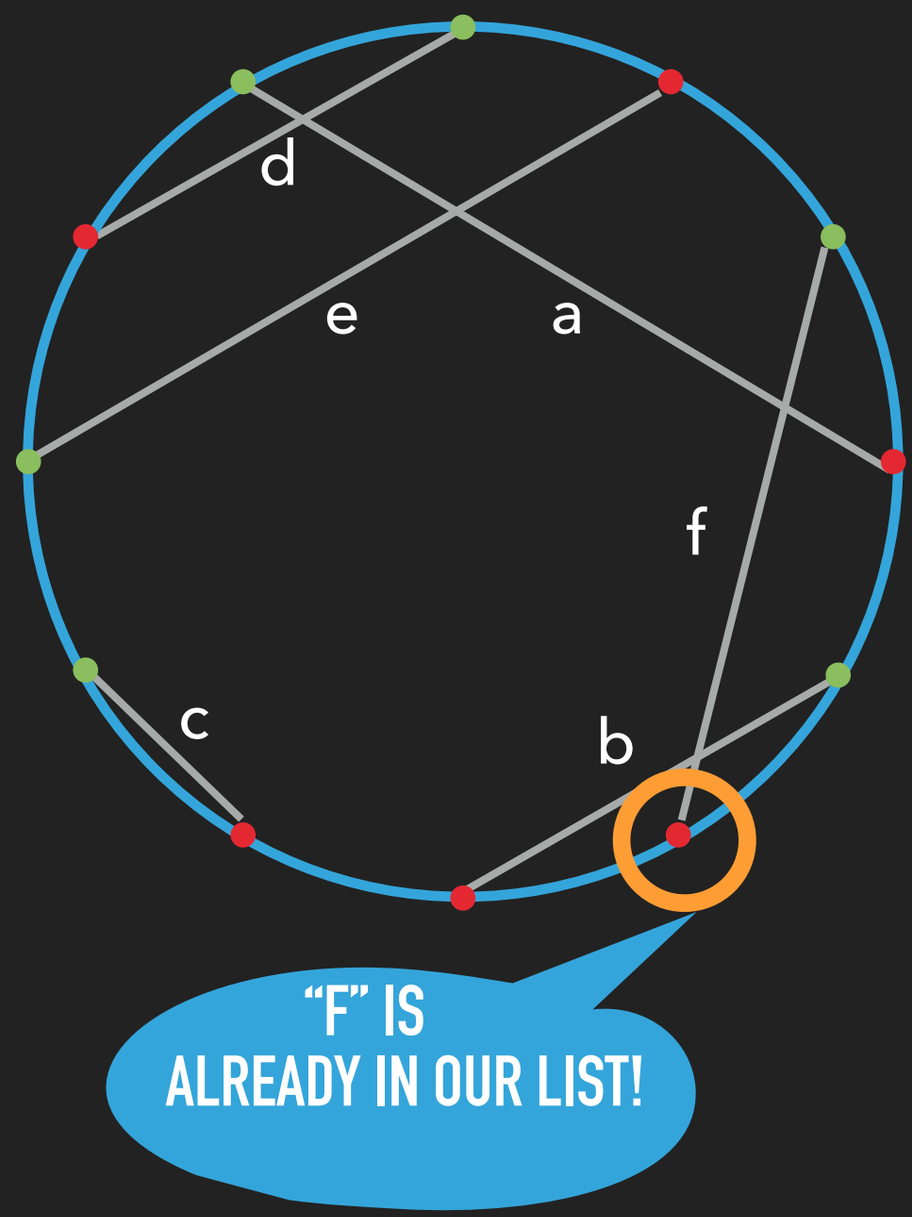
CIRCLE INTERSECTIONS



TOTAL INTERSECTIONS — 3

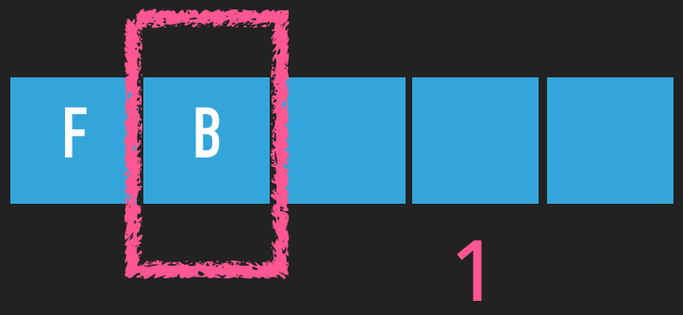
CIRCLE INTERSECTIONS

HOW MANY LINE SEGMENTS BEGAN AFTER “F” THAT HAVEN’T YET ENDED?

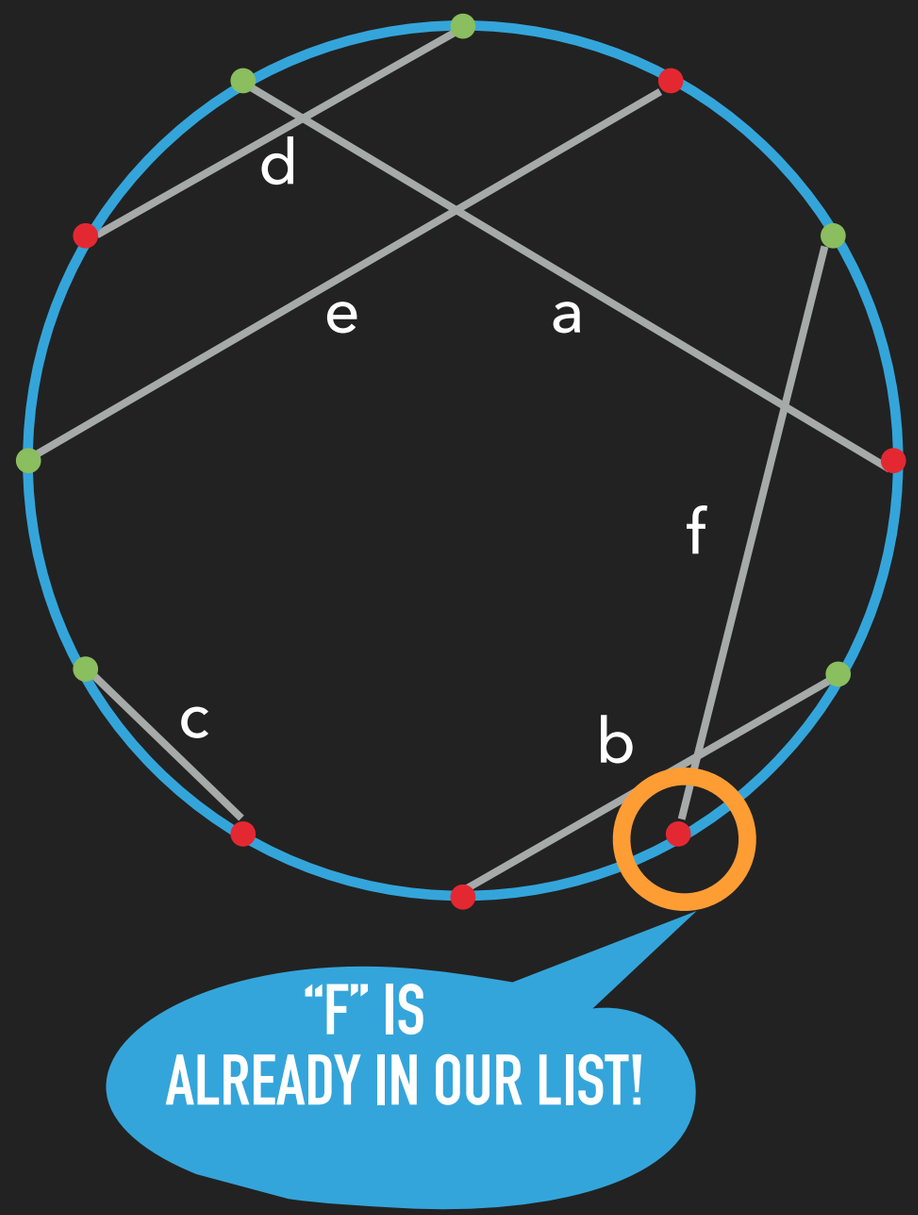


TOTAL INTERSECTIONS — 3

CIRCLE INTERSECTIONS

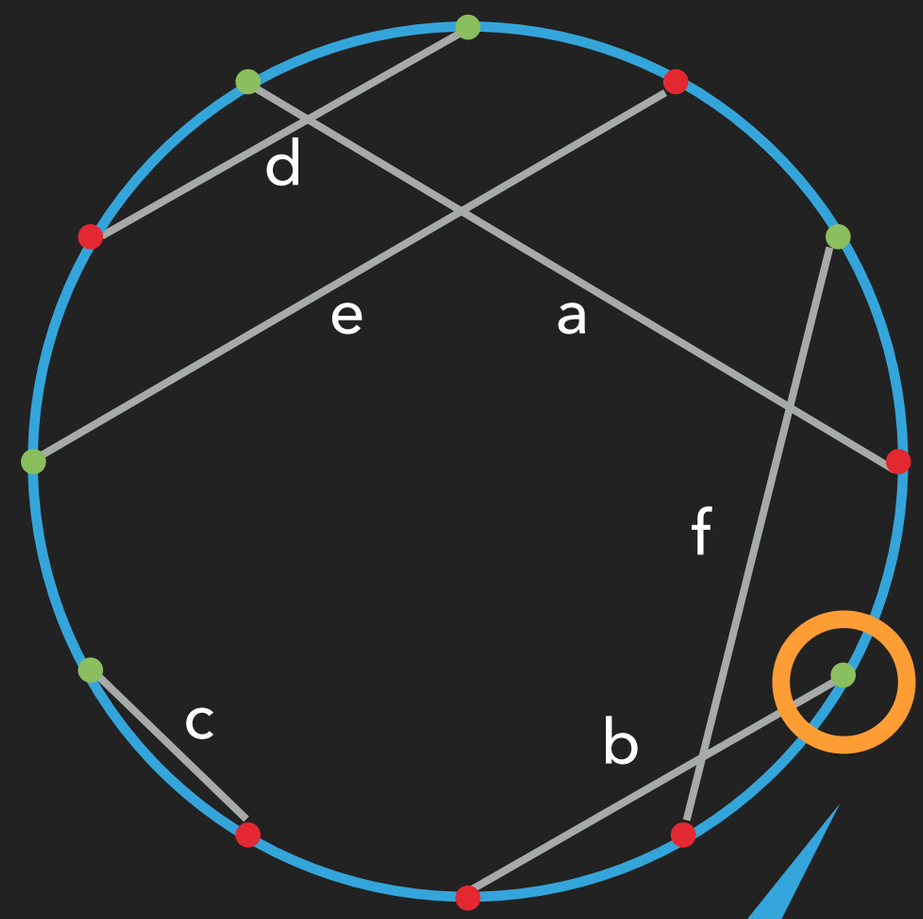


HOW MANY LINE SEGMENTS BEGAN AFTER “F” THAT HAVEN’T YET ENDED?



TOTAL INTERSECTIONS — 4

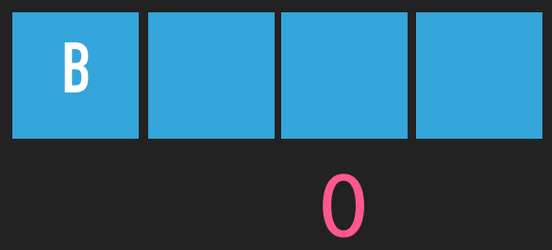
CIRCLE INTERSECTIONS



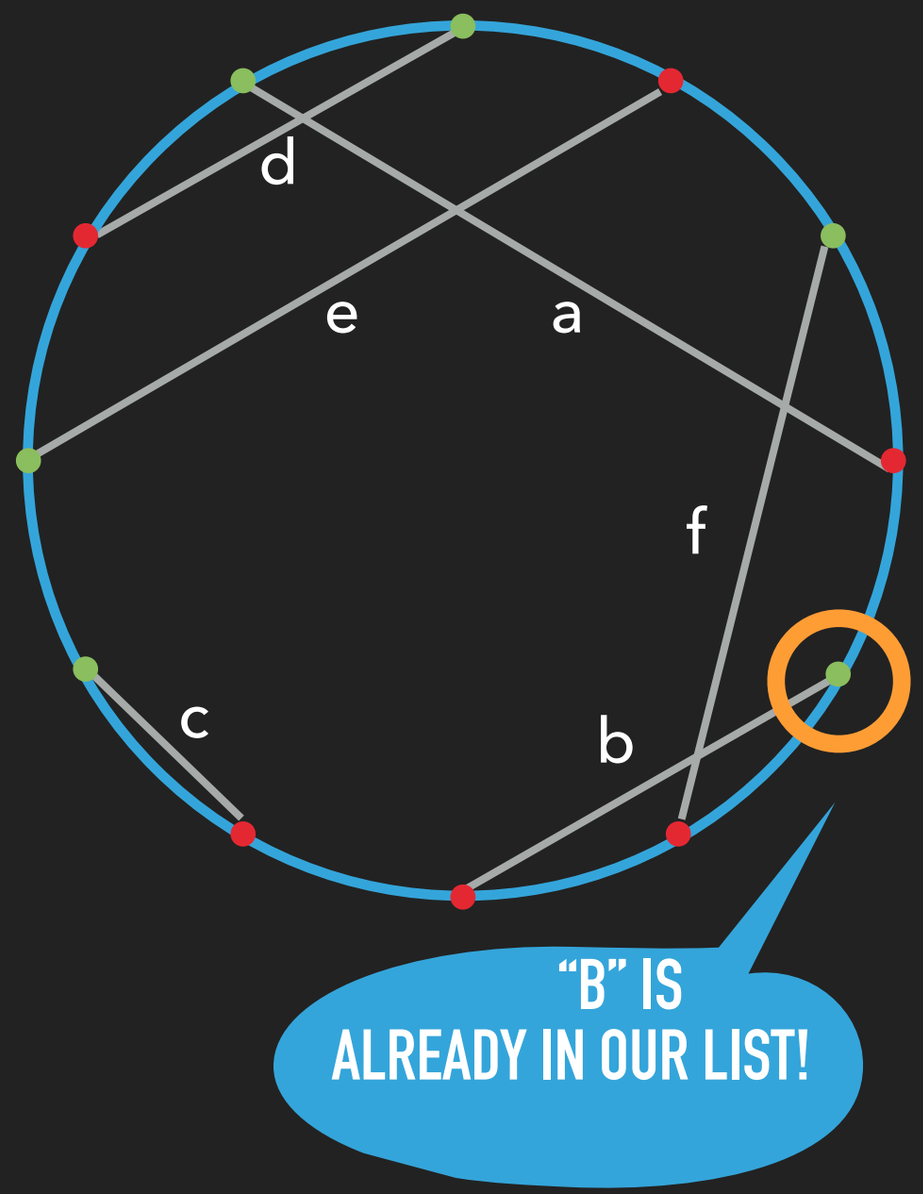
“B” IS
ALREADY IN OUR LIST!

TOTAL INTERSECTIONS — 4

CIRCLE INTERSECTIONS

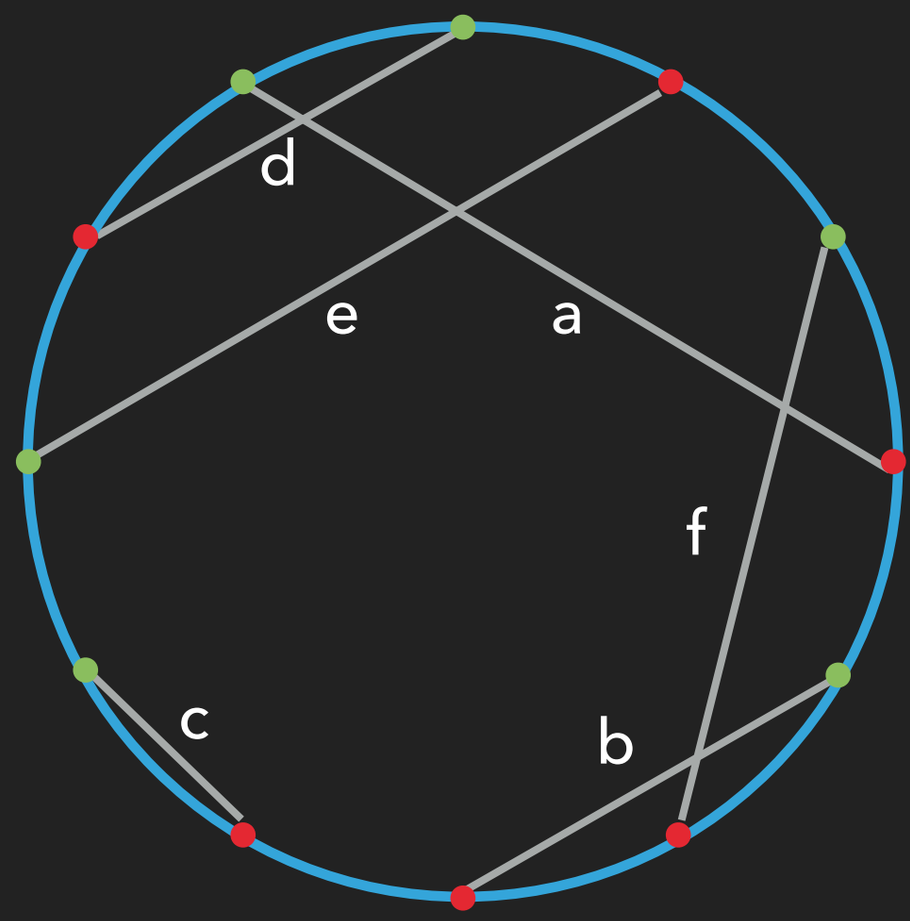


HOW MANY LINE SEGMENTS BEGAN AFTER “F” THAT HAVEN’T YET ENDED?



TOTAL INTERSECTIONS — 4

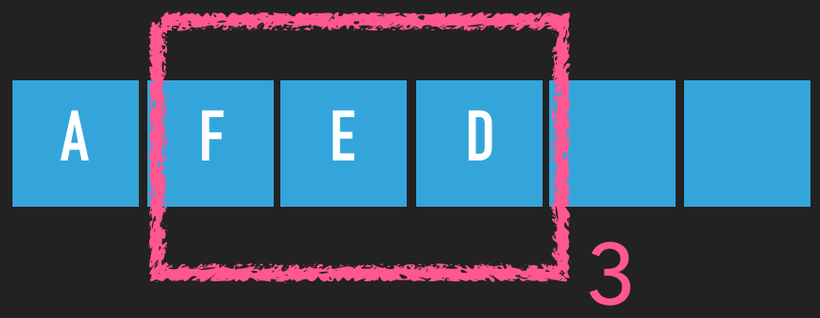
CIRCLE INTERSECTIONS



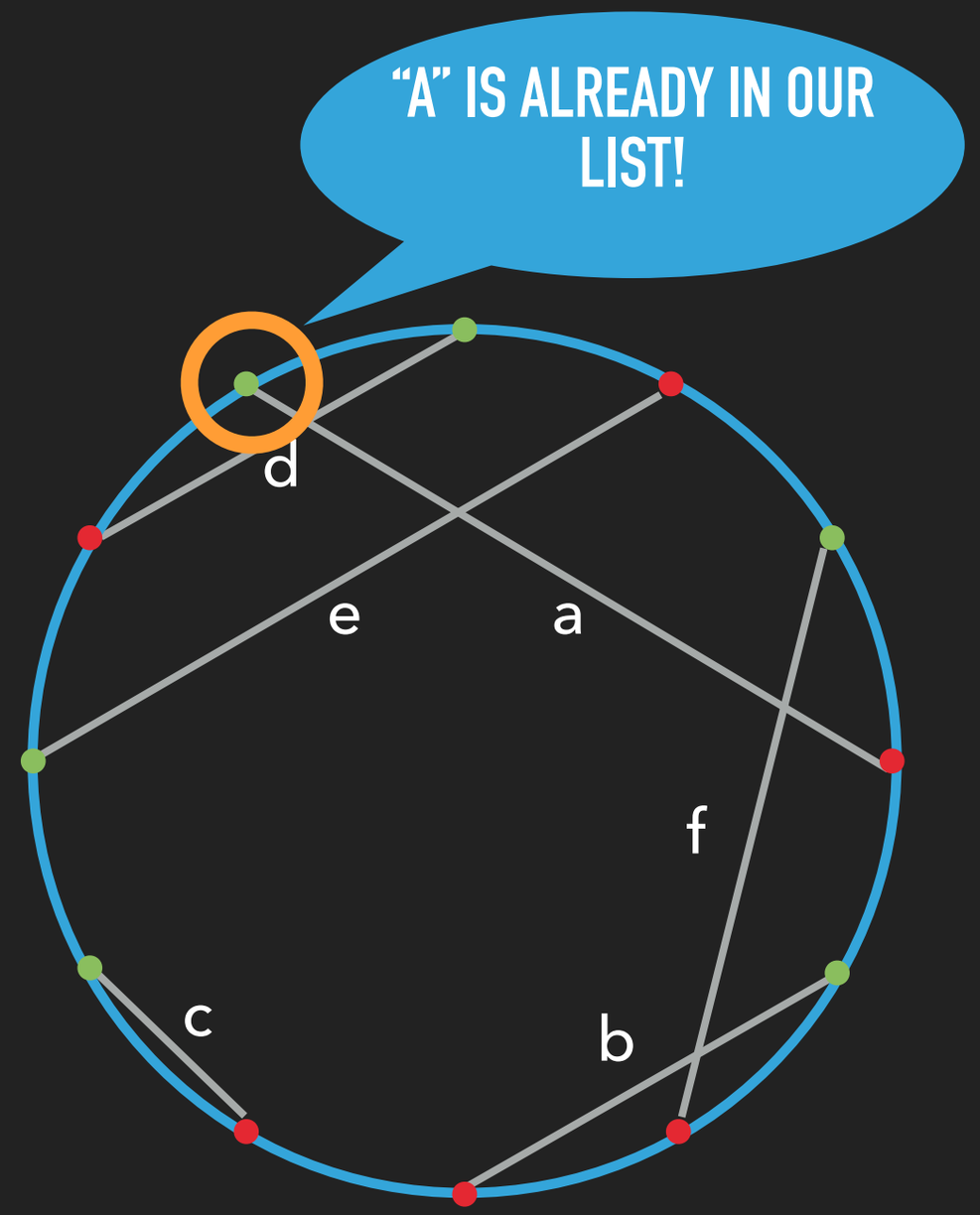
ALL DONE!

TOTAL INTERSECTIONS — 4

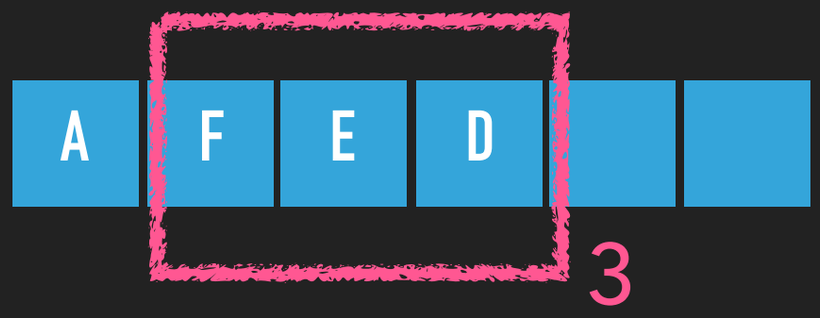
CIRCLE INTERSECTIONS



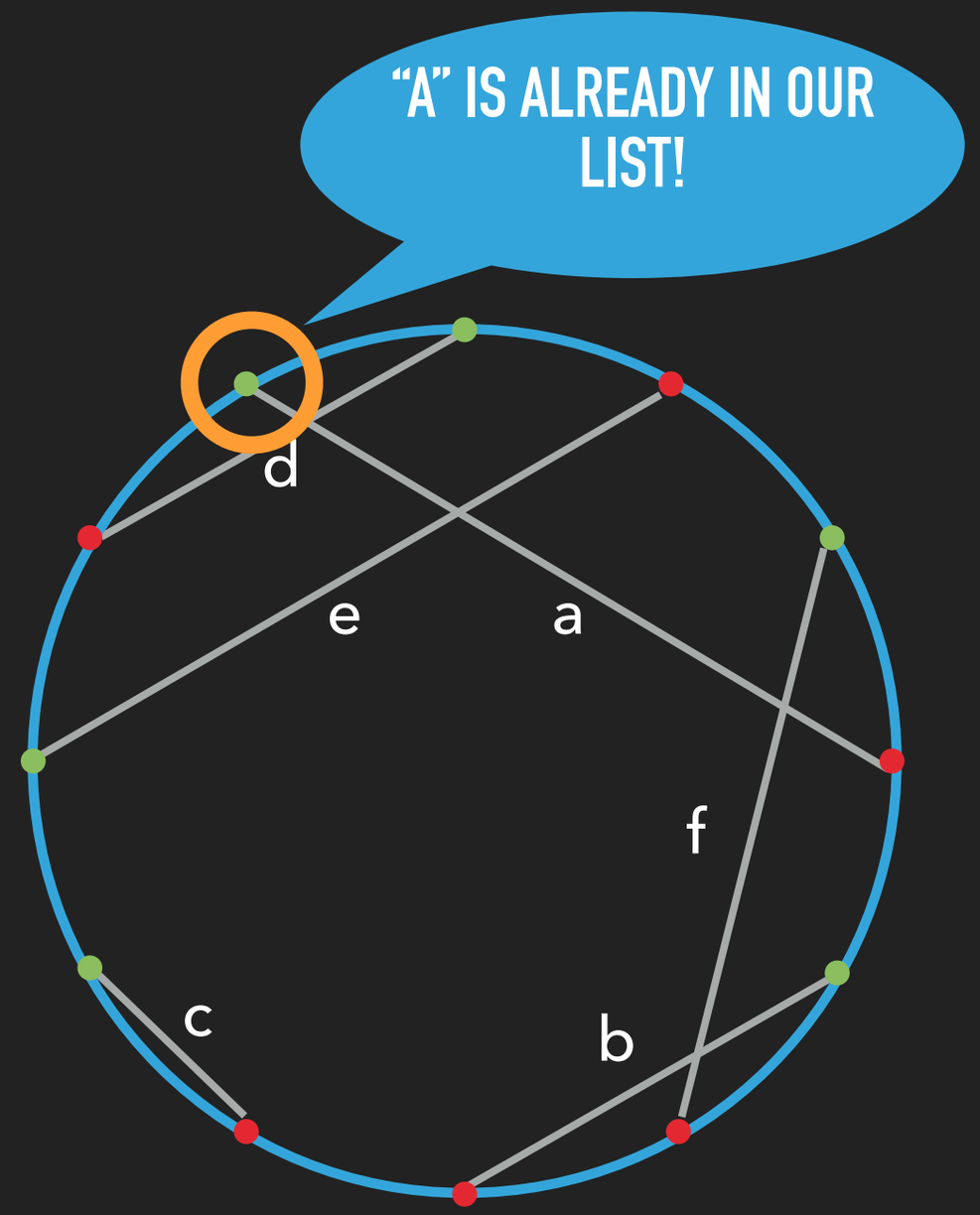
HOW MANY LINE SEGMENTS BEGAN AFTER "A" THAT HAVEN'T YET ENDED? (WHAT'S THE GENERAL ANSWER?)



CIRCLE INTERSECTIONS

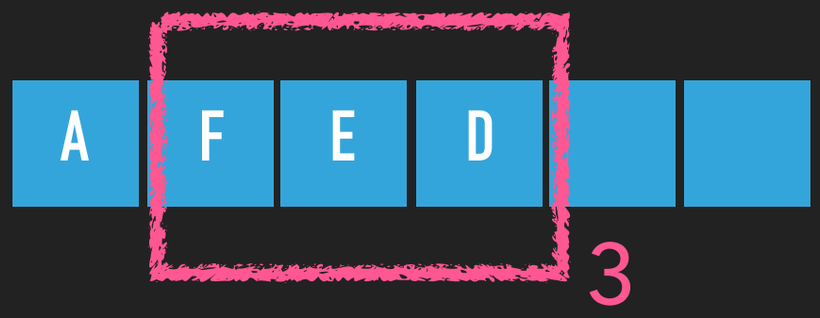


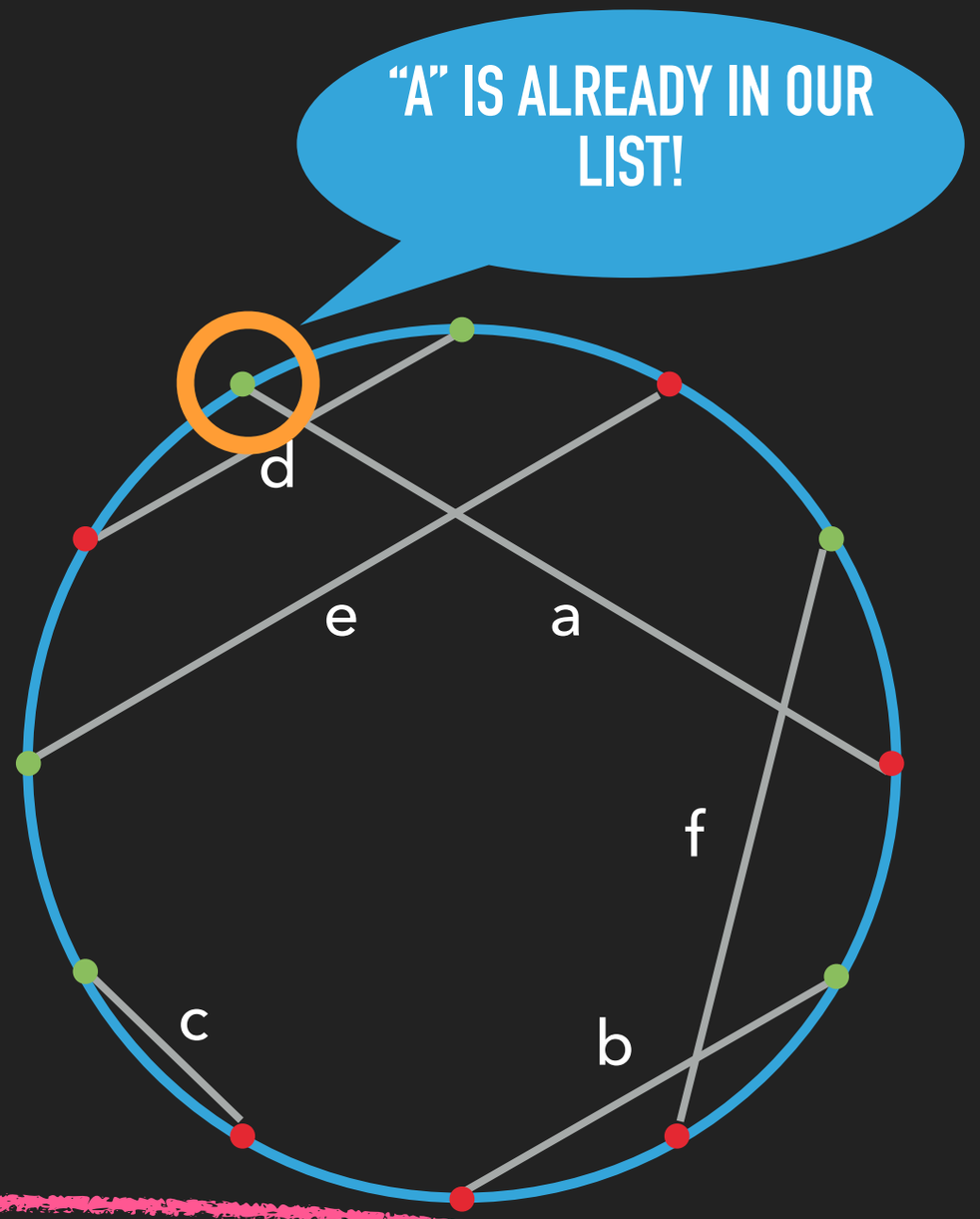
HOW MANY LINE SEGMENTS BEGAN AFTER "A" THAT HAVEN'T YET ENDED? (WHAT'S THE GENERAL ANSWER?)



n = list size - position of the line segment in sorted list

CIRCLE INTERSECTIONS





HOW MANY LINE SEGMENTS BEGAN AFTER "A" THAT HAVEN'T YET ENDED? (WHAT'S THE GENERAL ANSWER?)

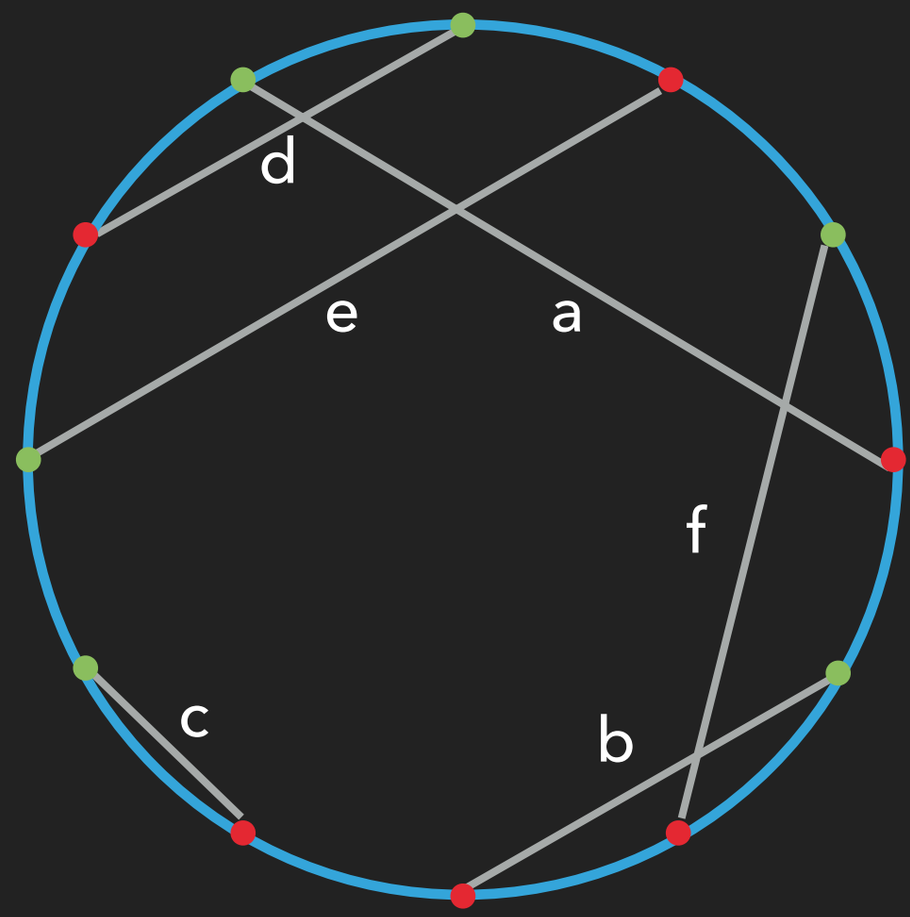
n = list size - position of the line segment in sorted list

Rank!

CIRCLE INTERSECTIONS

We'll keep a list sorted by angle.

Can be realized using an augmented BBST!



CIRCLE INTERSECTIONS (OPERATIONS)

- Lookup if the line segment is already in the BBST.
- Get the rank of an element.
- Add an element.
- Delete an element.
- ✓ $O(\log n)$ for all operations!