```
1: def example(lst): # n is len(lst)
2:     result = []
3:     for i in range(0, len(lst), 2):
4:         if lst[i] != lst[i+1]:
5:             average = (lst[i] + lst[i+1]) / 2
6:             if average in lst:
7:                 result.append(average)
8:     return count
```

Look at slide #38 in Lecture 7-1: Runtime and Big-O Notation!!

Line 3: iterates n/2 times; multiply that by work done by loop body

Line 4: conditional with a constant check, add it to rest of loop body.

Line 6: conditional with a O(n) check , add n to rest of body.

Lines 2, 5, 7, and 8 don't depend on size of input; they're constant.

What is the runtime of each of the following lines of code?

A. Line 3: O(N)/linear
B. Line 4: O(1)/constant
C. Line 5: O(1)/constant
D. Line 6: O(N)/linear
E. Line 7: O(1)/constant

What is the total runtime of the above code in Big-O?

Total: O(n^2)

Runtime: constant + n/2 * (constant + constant + n + constant) = constant + constant * (n2) + constant * n = O(n2)