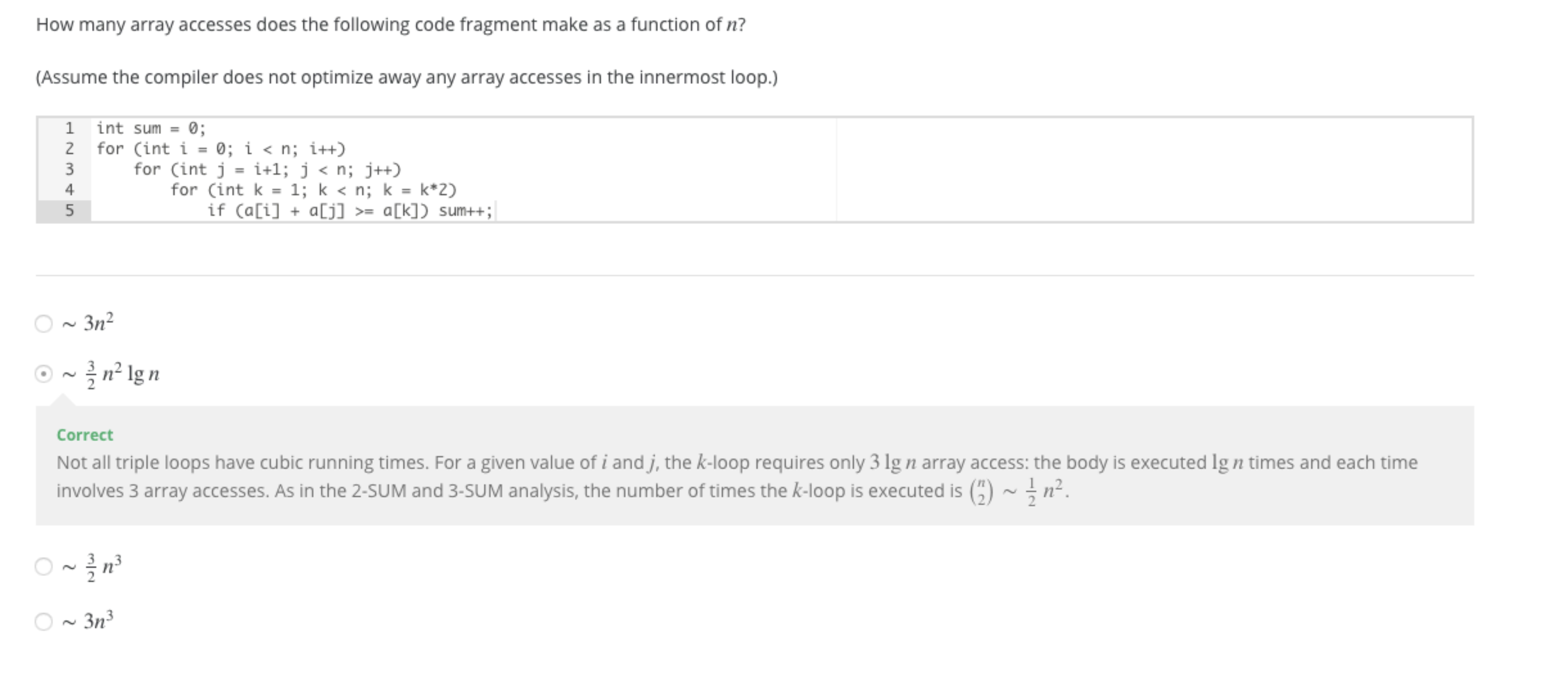

How many array accesses does the following code fragment make as a function of n?

(Assume the compiler does not optimize away any array accesses in the innermost loop.)

```
int sum = 0;
for (int i = 0; i < n; i++)
    for (int j = i+1; j < n; j++)
        for (int k = 1; k < n; k = k*2)
            if (a[i] + a[j] >= a[k]) sum++;
```

○ $\sim 3n^2$

◉ $\sim \frac{3}{2} n^2 \lg n$

Correct

Not all triple loops have cubic running times. For a given value of i and j, the k-loop requires only $3 \lg n$ array access: the body is executed $\lg n$ times and each time involves 3 array accesses. As in the 2-SUM and 3-SUM analysis, the number of times the k-loop is executed is $\binom{n}{2} \sim \frac{1}{2} n^2$.

○ $\sim \frac{3}{2} n^3$

○ $\sim 3n^3$