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.)

- ○ $\sim 3N^2$
- ◉ $\sim \frac{3}{2} N^2 \lg N$
- ○ $\sim \frac{3}{2} N^3$
- ○ $\sim 3N^3$

```
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++;
```

Loop 1

i ⇒ 0, 1, 2, 3 ..... N-1 { N times }

Loop 2

j ⇒ {1, 2, ... N-1}, {2, 3, ... N-1}, {3, 4 ...}, ...

Count of the Loop 2

= (N-1) + (N-2) + (N-3) + ... + 0

$= \frac{(N)(N-1)}{2}$ (2nd loop runs $\frac{(N)(N-1)}{2}$ times)

Loop 3

k ⇒ 1, 2, 4, 8 .....

If N = 10, k ⇒ 1, 2, 4, 8.

N = 33 k ⇒ 1, 2, 4, 8, 16, 32 $\{\log_2(33-1)\}+1$

N = N k ⇒ 1, 2 .....

count of loop 3 = $\log_2(N-1)+1 \simeq \log_2(N)$ for large N.

Total order of growth = $\frac{N^2 \log N}{2}$, Array access = $(3N^2 \log N)/2$