Lets look at the code we are evaluating :

```
int i, j, k = 0;
  for (i  = n/2; i <= n; i++) {
      for (j = 2; j <= n; j = j * 2) {
          k = k + n/2;
      }
  }
```

Now, lets just assume `n = 8` for now.
We will try to see, the values of j corresponding to each i.

```
i = 4, j = 2, 4, 8
i = 5, j = 2, 4, 8
i = 6, j = 2, 4, 8
i = 7, j = 2, 4, 8
i = 8, j = 2, 4, 8
```

If you notice, j keeps doubling till it is less than or equal to n. Number of times, you can double a number till it is less than n would be log(n).

Lets take more examples here to convince ourselves.

```
n = 16, j = 2, 4, 8, 16
n = 32, j = 2, 4, 8, 16, 32
```

So, j would run for O(log n) steps.
i runs for n/2 steps.

So, total steps ` = O (n/ 2 * log (n)) = O(n logn) `