

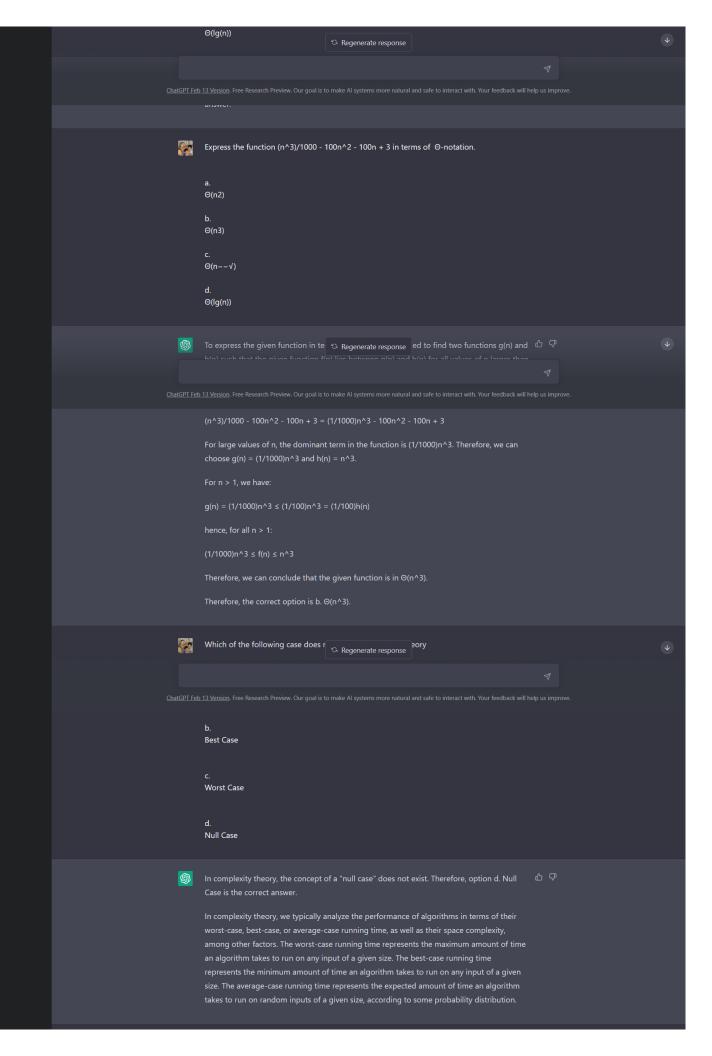
```
Rearranging the inequality, we get:
       For this inequality to hold for all n \ge n0, we must have (7/c) \le 1, since the coefficient of n on
       In other words, there are infinitely many constants c that satisfy the condition, and the
       The worst case complexity of Bubble sort algorithm is
       O(n2)
                                            S Regenerate response
       O(n)
       O(n log(n))
       This expression is O(n^2) in terms \bigcirc Regenerate response its, and hence the worst-case
What is the time complexity of the following code?
       for (i = N / 2; i \le N; i++) {
       O(N*Sqrt(N))
       O(N*log(N))
       O(N*N)
       log(N) times (base 2). Therefore, the total number of times the inner loop is executed is:
```

```
log(24...*N) = log(2^{(log(N/2))} * (N/2)!) = (log(N/2) + 1) * (N/2)
The time complexity of the code is therefore O(N log(N)) since the total number of iterations
of the inner loop is proportional to N log(N).
No. of elements
in the array (N) Execution time
(micro seconds)
 2048 202
 4096 805
 16384 12900
32768 51592
What can be the possible average case time complexity of this sorting algorithm?
Select one:
O(4N)
O(Nlg(N))
                                     S Regenerate response
O(N)
execution time against the input size on a log-log scale, and check if the plot looks
approximately linear.
\log(T(N)) = k + c * \log(N)
complexity exponent.
Plotting the log(T(N)) values again:

So Regenerate response straight line on the log-log scale, ii.

Plotting the log(T(N)) values again:

Regenerate response symplexity of the sorting
algorithm is O(N^c) = O(N^1.99), which is closer to O(N^2) than O(N \log(N)).
Express the function in terms of \Theta-notation.
Θ(n2)
Θ(n3)
Θ(n--√)
```



S Regenerate response