**Question 1:** Algorithm in which best-case time is equal to worst-case time:

Ans: Reverse an array the best case and worst-case time complexity is always same

**Question 2.** Order them based on their complexity.

2^N

2^(n+1)

2^2n

2^2^n

**Question 3.**  Mention one algorithm you know for each of the time complexities listed.

O (1), O (log n), O(n), O(n log n), O(n2), O(n3), O(2n)

O (1): Indexing an array

O (log n): Binary Search

O(n): Finding Max in an unsorted array

O (n log n): Merge Sort

O(n^2): Bubble Sort

O(n^3): Three Variable equation solver

O(2^n): Fibonacci with recursion

**Question 4.**  Apply Master Theorem and determine the time complexity of

fib(n) shown in slide 48.



**Question 5**

Solve the recurrence

T (1) = 1

T(n) = 2T(n/2) + c

Hint: Assume n = 2^k.

Text, letter

Description automatically generatedText, letter

Description automatically generated