|  |  |  |
| --- | --- | --- |
| **Input Length** | **Worst Accepted Time Complexity** | **Usually type of solutions** |
| 10 -12 | O(N!) | [Recursion](https://www.geeksforgeeks.org/recursion/) and [backtracking](https://www.geeksforgeeks.org/backtracking-algorithms/) |
| 15-18 | O(2N\* N) | Recursion, backtracking, and[bit manipulation](https://www.geeksforgeeks.org/bits-manipulation-important-tactics/) |
| 18-22 | O(2N\* N) | Recursion, backtracking, and bit manipulation |
| 30-40 | O(2N/2\* N) | [Meet in the middle](https://www.geeksforgeeks.org/meet-in-the-middle/), [Divide and Conquer](https://www.geeksforgeeks.org/divide-and-conquer-introduction/) |
| 100 | O(N4) | [Dynamic programming](https://www.geeksforgeeks.org/dynamic-programming/), [Constructive](https://www.geeksforgeeks.org/basic/constructive-algorithms/) |
| 400 | O(N3) | Dynamic programming, Constructive |
| 2K | O(N2\* log N) | Dynamic programming, [Binary Search](https://www.geeksforgeeks.org/binary-search/),[Sorting](https://www.geeksforgeeks.org/sorting-algorithms/),  Divide and Conquer |
| 10K | O(N2) | Dynamic programming, [Graph](https://www.geeksforgeeks.org/graph-data-structure-and-algorithms/), [Trees](https://www.geeksforgeeks.org/binary-tree-data-structure/), Constructive |
| 1M | O(N\* log N) | Sorting, Binary Search, Divide and Conquer |
| 100M | O(N), O(log N), O(1) | Constructive, [Mathematical,](https://www.geeksforgeeks.org/mathematical-algorithms/) [Greedy Algorithms](https://www.geeksforgeeks.org/greedy-algorithms-general-structure-and-applications/) |