**MUST-DO ALGORITHMS for CODING ROUNDS**

(Only to be done 3-6 months before placements)

(\*\* Will be enough for Amazon, Microsoft and similar companies coding rounds)

(\*\* Won’t be enough for Codenation, Directi level companies)

1.Binary Search

* <https://codeforces.com/problemset/problem/1354/B> (Easy)
* <https://www.interviewbit.com/problems/allocate-books/> (Medium)
* <https://codeforces.com/problemset/problem/1359/C>   
  (Hard -> no need to do if very less time is left)

2. Prefix Sum

* <https://cses.fi/problemset/task/1646> (easy)
* <https://www.hackerrank.com/contests/ab-yeh-kar-ke-dikhao/challenges/kj-and-street-lights/problem> (Medium -> Scanline Algo)
* <https://www.codechef.com/CENS2020/problems/CENS20A> (Hard)

3. Primes/Divisors

* <https://www.codechef.com/problems/CNTPRIME> (Easy)(Sieve)
* <https://www.spoj.com/problems/PRIME1/> (Medium) (Segmented Sieve)
* <https://cses.fi/problemset/task/2182> (hard -> can be left)

4. Divide and Conquer

* <https://www.spoj.com/problems/INVCNT/> (Easy)
* <https://cses.fi/problemset/task/1628> (Medium)
* <https://lightoj.com/problem/funny-knapsack> (Hard -> can be left)

5. String Algorithms

* <https://cses.fi/problemset/task/1753> (Easy) (KMP, Z, Rabin-Karp) (Solve using all 3 algos)
* <https://cses.fi/problemset/task/1111> (Medium)
* [https://codeforces.com/problemset/problem/271/D](https://codeforces.com/problemset/problem/271/D?locale=en) (Medium/Hard)

6. Tree Algorithms

* <https://cses.fi/problemset/task/1674> (Easy)
* <https://cses.fi/problemset/task/1131> (Medium)
* <https://cses.fi/problemset/task/1135> (Hard, covers LCA using Binary Lifting)

7. Graph Algorithms

* BFS Questions super duper important (<https://cses.fi/problemset/task/1192>)

(Also do problems like <https://cses.fi/problemset/task/1193> )

* <https://cses.fi/problemset/task/1671> (Dijsktra)
* <https://www.spoj.com/problems/EC_P/> (Bridges)
* <https://www.spoj.com/problems/SUBMERGE/> (Articulation Point)
* Rest do all Graph problems from Striver’s Graph series (<https://www.youtube.com/watch?v=YTtpfjGlH2M&list=PLgUwDviBIf0rGEWe64KWas0Nryn7SCRWw>)

8. Disjoint Set

* <https://www.hackerearth.com/practice/data-structures/disjoint-data-strutures/basics-of-disjoint-data-structures/practice-problems/algorithm/disjoint-set-union/>

<https://www.youtube.com/watch?v=3gbO7FDYNFQ&t=11s>

* <https://codeforces.com/contest/25/problem/D> (Medium)
* <https://www.spoj.com/problems/CLFLARR/> (Hard -> offline solution)

9. Segment Trees

* <https://cses.fi/problemset/task/1647> (Simple range query) (<https://www.youtube.com/watch?v=-dUiRtJ8ot0>)
* <https://cses.fi/problemset/task/1649> (Range query with point update)

(<https://www.youtube.com/watch?v=-dUiRtJ8ot0>)

* <https://cses.fi/problemset/task/1735> (hard-> can be left .. ) (<https://www.youtube.com/watch?v=rwXVCELcrqU>)

10. Dynamic Programming

* Generally the problems are variations of standard DP problems in geeksforgeeks. Do the problems named as “DP-3” to DP-28” on GFG, will automatically be covered if you doing SDE sheet)
* Digit DP (hard -> might appear if you are giving rounds in Hackerearth, else will not..)

<https://cses.fi/problemset/task/2220>