

**CP List(Solely for preparing for Coding Rounds of Top Prod
Based Companies and to do well in Coding Sites**

SDE SHEET: https://bit.ly/takeUforward_SDE

Linear Search:

1. <https://www.hackerearth.com/practice/algorithms/searching/linear-search/practice-problems/algorithm/simple-search-4/>
2. <https://www.hackerearth.com/practice/algorithms/searching/linear-search/practice-problems/algorithm/maximum-sum-4-f8d12458/>
3. <https://www.hackerearth.com/practice/algorithms/searching/linear-search/practice-problems/algorithm/mannas-first-name-4/>
4. <https://www.codechef.com/problems/SEGM01>
5. <https://www.hackerearth.com/practice/algorithms/searching/linear-search/practice-problems/algorithm/rest-in-peace-21-1/>

Hashing: (Basic and not String Hashing)

1. <https://www.hackerearth.com/practice/data-structures/hash-tables/basics-of-hash-tables/practice-problems/algorithm/maximum-occurrence-9/>
2. <https://codeforces.com/problemset/problem/4/C>
3. <https://www.hackerearth.com/practice/data-structures/hash-tables/basics-of-hash-tables/practice-problems/algorithm/perfect-pair-df920e90/description/>
4. <https://codeforces.com/problemset/problem/486/B>

PrefixSum:

1. <https://www.spoj.com/problems/CSUMQ/>
2. <https://www.codechef.com/problems/BLONDIE>
3. https://onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&category=24&page=show_problem&problem=1474
4. <https://www.hackerrank.com/contests/ab-yeh-kar-ke-dikhao/challenges/kj-and-street-lights/leaderboard> (learn the scanline algo trick, probably from here <https://www.youtube.com/watch?v=TSUvGqRFlug> (timeStamp: 2:00))
5. <https://www.codechef.com/COW42020/problems/COW3E/> (2d prefix sum)
6. <https://www.codechef.com/problems/COWA19B>
7. <https://www.codechef.com/problems/MXPOWER>

Sliding Window:

1. <https://www.hackerrank.com/challenges/min-max-riddle/problem>
2. <https://codeforces.com/problemset/problem/363/B>
3. <https://www.codechef.com/problems/SHIVIGOD> (try to do using sliding window)
4. <https://www.codechef.com/problems/BDGFT>
5. <https://www.codechef.com/problems/ECAPR206>
6. <https://codeforces.com/problemset/problem/1341/B>
7. <https://www.codechef.com/problems/SUMPOWER> (Can be solved using prefix sum, but try to do without that by using $O(1)$ space)

Binary Search:

(make sure you watch STL of BS ->

<https://www.youtube.com/watch?v=edJ19qIL8WQ>)

1. <https://www.hackerearth.com/practice/algorithms/searching/binary-search/practice-problems/algorithm/bishu-and-soldiers/>
2. <https://www.spoj.com/problems/AGGRCOW/>

3. <https://www.interviewbit.com/problems/painters-partition-problem/>
4. <https://codeforces.com/problemset/problem/975/C>
5. <https://www.codechef.com/problems/DSTROY>
6. <https://codeforces.com/problemset/problem/812/C>
7. <https://codeforces.com/problemset/problem/363/D>
8. <https://www.codechef.com/problems/FAKEBS>

Prime, Sieve, Segmented Sieve, Prime Factorisation:

1. <https://www.spoj.com/problems/PRIME1/>
2. <https://www.spoj.com/problems/TDPRIMES/>
3. <https://www.hackerearth.com/practice/math/number-theory/basic-number-theory-2/practice-problems/algorithm/nearest-prime-a828361b/>
4. <https://www.hackerearth.com/practice/math/number-theory/basic-number-theory-2/practice-problems/algorithm/ashu-and-prime-factors-4/>
5. <https://codeforces.com/contest/776/problem/B>
6. <https://www.hackerearth.com/practice/math/number-theory/basic-number-theory-2/practice-problems/algorithm/b-prime-counting/description/>
7. <https://www.hackerearth.com/practice/math/number-theory/primality-tests/practice-problems/algorithm/bob-and-gems-f8226fbd/description/>
8. <https://codeforces.com/contest/546/problem/D>
9. <https://codeforces.com/problemset/problem/222/C>

Search Combinatorics problems and do by self, since it is not an algorithm, rather mathematics, whose pattern can never be understood ;)

Constructive Problems having swapping terms in it:

1. <https://codeforces.com/problemset/problem/1353/B>
2. <https://codeforces.com/problemset/problem/489/A>
3. <https://codeforces.com/problemset/problem/920/C>
4. <https://codeforces.com/problemset/problem/1215/C>
5. <https://www.codechef.com/problems/SWAPPALI>

Bit Manipulation/Power Set:

1. <https://www.hackerearth.com/practice/basic-programming/bit-manipulation/basics-of-bit-manipulation/practice-problems/algorithm/find-the-numbers-75f24949/>

2. <https://www.hackerearth.com/practice/basic-programming/bit-manipulation/basics-of-bit-manipulation/practice-problems/algorithm/power-of-2-6/>
3. <https://codeforces.com/problemset/problem/1095/C>
4. <https://codeforces.com/problemset/problem/1202/A>
5. <https://codeforces.com/problemset/problem/1152/B>
6. <https://codeforces.com/problemset/problem/611/B>
7. <https://codeforces.com/problemset/problem/1097/B> (Power Set use)
8. <https://codeforces.com/problemset/problem/276/D>

Greedy Algorithms (A topic in which you need to many many problems):

1. <https://codeforces.com/problemset/problem/1291/A>
2. <https://codeforces.com/problemset/problem/1375/B>
3. <https://codeforces.com/problemset/problem/1294/C>
4. <https://codeforces.com/problemset/problem/1285/B> (Kadane's Algo pre-req)
5. <https://codeforces.com/problemset/problem/1201/B>
6. <https://codeforces.com/problemset/problem/274/A>
7. <https://codeforces.com/problemset/problem/413/C>
8. <https://codeforces.com/problemset/problem/1368/B>
9. <https://codeforces.com/problemset/problem/1291/B>

Divide and Conquer:

1. <https://leetcode.com/problems/reverse-pairs/> (Check my video on YT)
2. <https://codeforces.com/problemset/problem/768/B>
3. <https://cses.fi/problemset/task/1628>
4. <https://codeforces.com/problemset/problem/1263/C> (try to solve using MIM)
5. <https://codeforces.com/problemset/problem/1249/C2>
6. <https://codeforces.com/problemset/problem/1373/D>

Stack/Queues/PriorityQueues:

1. <https://www.hackerrank.com/challenges/balanced-brackets/problem>
2. <https://www.codechef.com/status/THESA>
3. <https://www.spoj.com/problems/ANARC09A/>
4. <https://www.hackerearth.com/practice/data-structures/queues/basics-of-queues/practice-problems/algorithm/monk-and-power-of-time-3a648bf0/>
5. <https://www.hackerearth.com/challenges/competitive/code-monk-heaps-and-priority-queues-1/algorithm/little-monk-and-williamson/>
6. <https://codeforces.com/contest/5/problem/C>
7. <https://www.hackerearth.com/practice/data-structures/stacks/basics-of-stacks/practice-problems/algorithm/little-shino-and-pairs/>
8. <https://www.hackerearth.com/practice/data-structures/trees/heapspriority-queues/practice-problems/algorithm/seating-arrangement-6b8562ad/>

String Algorithms(Hashing, Rabin Karp, KMP, Z-Function, Manacher's Algo):

1. <http://codeforces.com/problemset/problem/271/D>
2. <https://www.spoj.com/problems/NHAY/>
3. <https://www.spoj.com/problems/NAJPF/>

4. https://onlinejudge.org/index.php?option=onlinejudge&page=show_problem&problem=396
5. <http://codeforces.com/problemset/problem/126/B>
6. <http://codeforces.com/problemset/problem/271/D>
7. <https://www.codechef.com/problems/RUNNING>
8. https://www.codechef.com/problems/INSQ15_A
9. <https://codeforces.com/problemset/problem/346/B>
10. <https://codeforces.com/problemset/problem/432/D> (Trees must be known)
11. <https://leetcode.com/problems/longest-palindromic-substring/> (Manacher's)
12. <https://leetcode.com/problems/longest-palindromic-substring/>
13. <https://codeforces.com/contest/1080/problem/E> (Super tough)

Tree's (DFS, LCA, Subtree size ..) :

1. <https://cses.fi/problemset/task/1674>
2. <https://cses.fi/problemset/task/1130>
3. <https://www.spoj.com/problems/ABCPATH/>
4. <https://cses.fi/problemset/task/1131>
5. <https://codeforces.com/problemset/problem/1336/A>
6. <https://codeforces.com/contest/734/problem/E> (Bit tougher DFS)
7. <https://cses.fi/problemset/task/1688> (LCA)
8. <https://www.spoj.com/problems/DISQUERY/>
9. <https://cses.fi/problemset/task/1131> (LCA)
10. <https://cses.fi/problemset/task/1135> (LCA)
11. <https://codeforces.com/contest/208/problem/E>
12. <https://codeforces.com/contest/1328/problem/E>
13. <https://codeforces.com/contest/519/problem/E>
14. Still want more for LCA, find here -> <https://codeforces.com/blog/entry/43917>

Graph Algorithms (DFS, BFS, Dijkstra, Floyd Washall, Bellman Ford, Bridges, 0-1 BFS, Bipartite, Topo-sort ...) :

1. <https://cses.fi/problemset/task/1192> (bfs)
2. <https://cses.fi/problemset/task/1193>
3. <https://codeforces.com/problemset/problem/242/C>
4. <https://cses.fi/problemset/task/1193> (Connected Components)
5. <https://cses.fi/problemset/task/1667>
6. <https://cses.fi/problemset/task/1669>
7. <https://cses.fi/problemset/task/1671> (Dijkstra)
8. <https://codeforces.com/problemset/problem/20/C>
9. <https://cses.fi/problemset/task/1672> (Floyd Warshall)
10. <https://cses.fi/problemset/task/1673>
11. <https://cses.fi/problemset/task/1197> (Bellman Ford)
12. <https://cses.fi/problemset/task/1679> (topo sort)
13. <https://codeforces.com/problemset/problem/510/C>
14. <https://codeforces.com/problemset/problem/59/E> (tough Dijkstra)
15. https://onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&page=sow_problem&problem=737
16. <https://www.spoj.com/problems/SUBMERGE/>

17. <https://www.codechef.com/problems/REVERSE> (0-1 BFS)
18. <https://codeforces.com/contest/1296/problem/E1> (Bipartite)

Once you have done this, if you feel like doing more, search and do as much as you can on the algo names above..

Dynamic Programming:

1. https://atcoder.jp/contests/dp/tasks/dp_a
2. https://atcoder.jp/contests/dp/tasks/dp_b
3. https://atcoder.jp/contests/dp/tasks/dp_c
4. https://atcoder.jp/contests/dp/tasks/dp_d
5. https://atcoder.jp/contests/dp/tasks/dp_e
6. https://atcoder.jp/contests/dp/tasks/dp_f
7. https://atcoder.jp/contests/dp/tasks/dp_h
8. https://atcoder.jp/contests/dp/tasks/dp_i
9. <https://cses.fi/problemset/task/1635>
10. <https://cses.fi/problemset/task/1636>
11. <https://codeforces.com/problemset/problem/1015/E1>
12. <https://codeforces.com/problemset/problem/977/F>
13. <https://codeforces.com/problemset/problem/1155/D>
14. <https://codeforces.com/problemset/problem/1341/D> (I also have a video on this, do check out)
15. <https://vjudge.net/problem/LightOJ-1068>
16. <https://vjudge.net/problem/LightOJ-1205>
17. <https://www.codechef.com/problems/DGTCNT>
18. <https://www.spoj.com/problems/CPCRC1C/>
19. <https://www.spoj.com/problems/PR003004/>
20. <https://codeforces.com/contest/628/problem/D>

Disjoint Set:

1. <https://www.hackerearth.com/practice/data-structures/disjoint-data-strutures/basics-of-disjoint-data-structures/practice-problems/algorithm/owl-fight/>
2. <https://www.hackerearth.com/practice/data-structures/disjoint-data-strutures/basics-of-disjoint-data-structures/practice-problems/algorithm/still-maximum/>
3. <https://codeforces.com/contest/25/problem/D>
4. <https://www.spoj.com/problems/CLFLARR/> (offline)
5. <https://codeforces.com/contest/151/problem/D>
6. <https://codeforces.com/problemset/problem/547/B>

Sqrt Decomposition:

1. <https://www.hackerearth.com/problem/algorithm/gcd-problem-1/>
2. <https://www.hackerearth.com/problem/algorithm/final-question/>
3. <https://codeforces.com/contest/220/problem/B>
4. <https://codeforces.com/contest/86/problem/D> (Mo's Algo)
5. <https://codeforces.com/contest/242/problem/E>
- 6.

Fenwick Tree:

1. <https://www.spoj.com/problems/INVCNT/>
2. <https://codeforces.com/gym/100741/problem/A>
3. <https://www.spoj.com/problems/MATSUM/>
4. <https://codeforces.com/gym/100741/problem/A>
5. <https://www.spoj.com/problems/DQUERY/>
6. <https://codeforces.com/problemset/problem/61/E>

Segment Tree(lazy also included):

1. <https://cses.fi/problemset/task/1646>
2. <https://cses.fi/problemset/task/1647>
3. <https://codeforces.com/problemset/problem/61/E>
4. <https://codeforces.com/contest/356/problem/A>
5. <https://codeforces.com/contest/459/problem/D>
6. <https://codeforces.com/contest/61/problem/E>
7. <https://codeforces.com/contest/380/problem/C>
8. <https://www.hackerearth.com/practice/data-structures/advanced-data-structures/fenwick-binary-indexed-trees/practice-problems/algorithm/help-ashu-1/>
9. <https://codeforces.com/contest/52/problem/C>
10. <https://codeforces.com/contest/52/problem/C>
11. <https://codeforces.com/contest/558/problem/E>
12. <https://codeforces.com/contest/558/problem/E>
13. <https://codeforces.com/contest/558/problem/E>