# Merge Sort

https://leetcode.com/problems/count-of-smaller-numbers-after-self/
(my solution:
https://leetcode.com/problems/count-of-smaller-numbers-after-self/discuss/1401108/c%2B%2B-or-merge-sort)
https://leetcode.com/problems/reverse-pairs/submissions/
(my solution: https://leetcode.com/submissions/detail/537871067/)
https://leetcode.com/problems/count-of-range-sum/
(my solution:
https://leetcode.com/problems/count-of-range-sum/discuss/1401342/c-merge-sort-easy-to-understand)

# DP

https://leetcode.com/problems/minimum-number-of-refueling-stops/
(Min. no. stops to reach the target)
(can be done using heaps in nlogn as well)
https://codeforces.com/contest/1526/problem/C1
(Based on inclusion exclusion principle)
https://www.codingninjas.com/codestudio/guided-paths/data-structures-algorithms/content/118824/offering/1382031?leftPanelTab=0
(Based on counting principle)
https://www.codingninjas.com/codestudio/guided-paths/data-structures-algorithms/content/118824/offering/1382009?leftPanelTab=0
(Based on optimizing tabulation)
https://leetcode.com/problems/constrained-subsequence-sum/
(Based on optimizing tabulation)
https://practice.geeksforgeeks.org/problems/matrix-chain-multiplication0303/1#
(based on partitioning of array)
https://leetcode.com/problems/subarray-sums-divisible-by-k/
(dp+map question)
https://youtu.be/LAKWWDX3sGw?t=885
(get string from dp)
https://leetcode.com/problems/shortest-common-supersequence/

(get string from dp)
https://leetcode.com/problems/burst-balloons/submissions/
(based on partition of array)
https://leetcode.com/problems/triples-with-bitwise-and-equal-to-zero/
(based on bitwise operation)
https://leetcode.com/problems/best-time-to-buy-and-sell-stock-with-transaction-fee/
(memoization and even dp is not sufficient here)
https://www.geeksforgeeks.org/longest-common-increasing-subsequence-lcs-lis/
(LCIS -> mugged up tabulation approach)
https://leetcode.com/problems/maximum-number-of-events-that-can-be-attended-ii/
(Important observation : can't be done using simple 0/1 knapsack)

https://leetcode.com/problems/make-array-strictly-increasing/

https://leetcode.com/problems/maximum-number-of-events-that-can-be-attended-ii/

(Important concept: Skip taking prev into account by making use of upper_bound)
https://leetcode.com/problems/jump-game-vi/
(optimizing dp)
https://leetcode.com/problems/number-of-sets-of-k-non-overlapping-line-segments/
(optimizing dp)
https://leetcode.com/problems/find-the-longest-valid-obstacle-course-at-each-position/discuss/1390603/Straightup-LIS
(LIS type)
https://leetcode.com/submissions/detail/536873709/
(My solution)

# Graphs

https://www.codingninjas.com/codestudio/guided-paths/data-structures-algorithms/content/118824/offering/1381994?leftPanelTab=1

-Dijkstra
https://leetcode.com/problems/path-with-minimum-effort/
https://leetcode.com/problems/cheapest-flights-within-k-stops/submissions/ (dp (ac) ; dijkstra (TLE))
https://leetcode.com/problems/network-delay-time/

# BFS

https://leetcode.com/problems/shortest-path-visiting-all-nodes/
(New type of traversal)
https://leetcode.com/problems/shortest-path-to-get-all-keys/
(My solution-
https://leetcode.com/problems/shortest-path-to-get-all-keys/discuss/1412847/c%2B%2B-or-intelligent-BFS-or-100-efficient-space-and-time)
(Intelligent bfs)

# Greedy

https://leetcode.com/problems/gas-station
(Starting gas station index to complete circular queue)
https://leetcode.com/problems/minimum-swaps-to-make-strings-equal/
(observation based)

# Arrays

https://leetcode.com/problems/next-greater-element-ii/discuss/98270/JavaC++Python-Loop-Twice
(Next greater - 2)
https://leetcode.com/problems/minimum-moves-to-equal-array-elements-ii/
(common problem)

https://leetcode.com/contest/weekly-contest-244/problems/minimum-number-of-flips-to-make-the-binary-string-alternating
(Based on making alternating [1010010 or 0101010] string problem)

https://leetcode.com/problems/minimum-moves-to-make-array-complementary
(New variety)
https://leetcode.com/problems/minimum-increment-to-make-array-unique/
(common and important)
https://leetcode.com/problems/minimum-number-of-operations-to-move-all-balls-to-each-box/
(common trick to reduce O(n2) to O(n))

# Stack

https://leetcode.com/problems/single-threaded-cpu/
(excellent use of stack)
https://leetcode.com/problems/non-overlapping-intervals/
(non overlapping intervals)
https://leetcode.com/problems/smallest-subsequence-of-distinct-characters/
(excellent use of stack)
https://leetcode.com/problems/verify-preorder-serialization-of-a-binary-tree/
(use this https://www.youtube.com/watch?v=_mbnPPHJmTQ)

Must do:

- 920 · Meeting Rooms - LintCode
- 919 · Meeting Rooms II - LintCode
- 1897 · Meeting Room III - LintCode
- 300 · Meeting Room IV - LintCode
- https://leetcode.com/problems/insert-interval/
- https://leetcode.com/problems/interval-list-intersections/

**String**

# Dequeue

https://leetcode.com/problems/shortest-subarray-with-sum-at-least-k/

# Sorting

https://leetcode.com/problems/reverse-pairs/
(i<j && a[i]>2*a[j])

# Binary search

https://leetcode.com/problems/4sum-ii/
https://leetcode.com/problems/minimum-interval-to-include-each-query/
https://leetcode.com/problems/find-a-value-of-a-mysterious-function-closest-to-target/discuss/743741/Detailed-General-Ideasolution-for-such-problems-(-AND-OR-GCD-)-in-O(N-*-log(-max(arri)-)-)

# Heaps

https://www.interviewbit.com/problems/maximum-sum-combinations/
(Maximum k sum pair)
Can be done from both heap and stack:

- 920 · Meeting Rooms - LintCode
- 919 · Meeting Rooms II - LintCode

# Hashing

https://leetcode.com/problems/flip-columns-for-maximum-number-of-equal-rows/
(New technique)
https://leetcode.com/problems/longest-well-performing-interval/(great qn)
https://leetcode.com/problems/triples-with-bitwise-and-equal-to-zero/
(bit manipulation)

# Bitmask

https://leetcode.com/problems/single-number-iii/
(new technique)
https://classroom.codingninjas.com/app/classroom/me/9738/content/165316/offering/2041989/problem/6162
(Problem 3 ->Unlucky number)

```cpp
#include<bits/stdc++.h>
using namespace std;
#define ll long long int

void solve(){
    ll n;
    cin>>n;
    n++;
    string ans;
    while(n){
        if(n%2) ans.push_back('3');
        else ans.push_back('1');
        n=(n>>1);
    }
    reverse(ans.begin(),ans.end());
    ans.erase(ans.begin());
    cout<<ans<<endl;
}

int main() {
    solve();
    return 0;
}
```

# Maths

https://leetcode.com/problems/reach-a-number/
https://leetcode.com/problems/broken-calculator/