# Codechef

### Find (subset all element xor) xor k is maximum

* 1. <https://www.codechef.com/problems/XORSUB>
  2. Dp[i][j] = 1 if only if there exists a subset P of A[1…i] such that Xor of all elements = j, otherwise 0
  3. for i = 1 to N:
  4. for j = 0 to 1023:
  5. dp[i][j] = dp[i - 1][j] | dp[i - 1][j ^ a[i]]
  6. Ans = maximum value of dp[n][j] \* (j ^ k) for all j.

### Minimum nested doll problem

* 1. <https://www.spoj.com/problems/MDOLLS/>
  2. Sort by dec order of width and when equ inc by height
  3. Add new doll to smallest ava height greater than current doll
  4. <https://journeywithdp.blogspot.com/2018/06/spoj-mdolls-nested-dolls.html>
  5. Similar easy question :
     1. <https://cses.fi/problemset/task/1073/>
     2. <https://www.spoj.com/submit/MSTICK/>

### Max absolute difference between two subarray

* 1. https://www.codechef.com/problems/DELISH
  2. For i in 0 to n-1
  3. Ans = max(leftMax[ i ] - rightMin[ i + 1 ], rightMax[ i + 1 ] - leftMin[ i ]
  4. Min and max array using caddance

# LeetCode

### Represent no with minimum no of Perfect Squares

* 1. <https://leetcode.com/problems/perfect-squares/>
  2. dp[i] = min(dp[i], dp[i-j\*j]+1); j >= 1 an i - j\*j >=0
  3. Using BFS

### Predict the winner

* 1. <https://leetcode.com/problems/predict-the-winner/>
  2. <https://www.geeksforgeeks.org/optimal-strategy-for-a-game-dp-31/>

| F(i, j) represents the maximum value the user can collect from i'th coin to j'th coin.  F(i, j) = Max(Vi + min(F(i+2, j), F(i+1, j-1) ),   Vj + min(F(i+1, j-1), F(i, j-2) )) As user wants to maximise the number of coins. ss  Base Cases  F(i, j) = Vi If j == i  F(i, j) = max(Vi, Vj) If j == i + 1 |
| --- |

### Partition array into two parts where both have equal sum

* 1. sum(subset1) + sum(subset2) = total\_sum
  2. sum(subset1) = total\_sum / 2
  3. So is there any subset having sum = total\_sum/2 ----> subset sum problem

### Partition array into k equal subsets

* 1. <https://leetcode.com/problems/partition-to-k-equal-sum-subsets/>

### Super ugly number

* 1. <https://leetcode.com/problems/super-ugly-number/>
  2. Maintain k pointers which pointes to last ugly number generated by kth prime number multiplication

### Largest subset sum divisible by k

* 1. <https://leetcode.com/problems/greatest-sum-divisible-by-three/>

|  | 0 | 1 | 2 |
| --- | --- | --- | --- |
| 3 | 3 | 0 | 0 |
| 6 | 9 | 0 | 0 |
| 5 | 9 | 0 | 14 |
| 1 | 15 | 10 | 14 |

### Burst Balloon Problem

* 1. <https://leetcode.com/problems/burst-balloons/submissions/>
  2. [https://leetcode.com/problems/burst-balloons/discuss/76229/For-anyone-that-is-still-confused-after-reading-all-kinds-of-explanations…](https://leetcode.com/problems/burst-balloons/discuss/76229/For-anyone-that-is-still-confused-after-reading-all-kinds-of-explanations%E2%80%A6)
  3. for (int k = left; k <= right; ++k)

dp[left][right] = max(dp[left][right], nums[left-1] \* nums[k] \* nums[right+1] + dp[left][k-1] + dp[k+1][right])\*\*

Updated

### Wiggle subseq

* 1. <https://leetcode.com/problems/wiggle-subsequence/>

### Split array into largest sum

### <https://leetcode.com/problems/split-array-largest-sum/>

### DP patterns https://leetcode.com/discuss/general-discussion/458695/Dynamic-Programming-Patterns

Atcoder

1. Find longest path in directed acyclic graph
   1. <https://atcoder.jp/contests/dp/tasks/dp_g>
   2. Find topological sorting
   3. For each u in topological sorting

If dp[u] + 1 > dp[v]

dp[v] = dp[u] + 1

# Matrix

### Maximum Square / Count Square in 2 D matrix

* 1. if(matrix[i][j] == 1) dp[i][j] = min(dp[i - 1][j - 1], min(dp[i - 1][j], dp[i][j - 1])) + 1
  2. <https://leetcode.com/problems/maximal-square/>
  3. <https://leetcode.com/problems/count-square-submatrices-with-all-ones/>

### Sub matrix with given sum

* 1. <https://www.interviewbit.com/problems/sub-matrices-with-sum-zero/>

For each row, calculate the prefix sum.

For each pair of columns,

calculate the accumulated sum of rows.

Now this problem is same to, "Find the Subarray with Target Sum".

* 1. [https://leetcode.com/problems/max-sum-of-rectangle-no-larger-than-k](https://leetcode.com/problems/max-sum-of-rectangle-no-larger-than-k/)

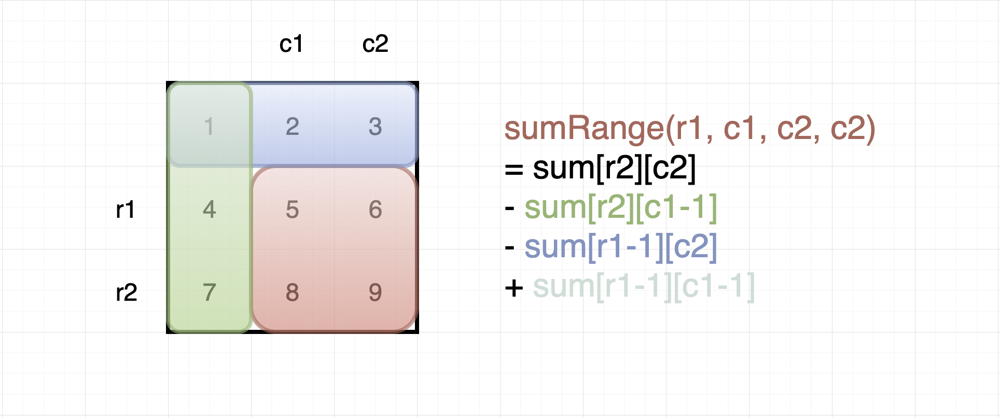
### Maximum Rectangle Area

* 1. <https://leetcode.com/problems/largest-rectangle-in-histogram>
  2. <https://leetcode.com/problems/maximal-rectangle/>
  3. Apply max histogram problem in each row and keep trace of maximum area

### Count all Rectangle

### <https://leetcode.com/problems/count-submatrices-with-all-ones/submissions/>

### <https://leetcode.com/problems/matrix-block-sum/>



Generate all possible squares

(Only Iterate for Square)

for(int i= 1; i < r; i++)

for(int j= 1;j < c; j++)

Int len = 0;

while(i - len >0 && j - len > 0)

R2 = i , R1=i-len , c2= j , c1= j-len  
 /// prefix   
 Len++;

### <https://leetcode.com/problems/odd-even-jump/>

### <https://leetcode.com/problems/number-of-ways-to-paint-n-3-grid/>

### <https://leetcode.com/problems/stone-game-v/>

### <https://www.interviewbit.com/problems/merge-elements/>