**Important info**

* Sort array in descending order:
  + sort(ara, ara+n, greater<int>());
* Sort vector in descending order:
  + sort(vt.begin(), vt.end(), greater<int>());
* Priority queue in ascending order:
  + priority\_queue< int, vector<int>, greater<int> > pq;
* Erasing element from set and iterating: (Erasing element invalidates the iterator. So, increment it before deleting the element.)
  + cur=maybe.begin();
  + while(cur!=maybe.end()){
  + if(!visit[\*cur]){
  + maybe.erase(cur++);
  + }
  + else cur++;
  + }
* Assigning 2d DP using vector
  + **vector**<**vector<int>** > dp;
  + dp.assign(m+1, **vector<int>**(n+1, -1));
* For 256MB(228 Byte) of memory int(22 Byte) array of size 226 (Roughly 6\*107) can be declared.
* Removing duplicate elements from a vector:
  + sort(point.begin(), point.end());
  + point.erase(unique(point.begin(), point.end()), point.end());

**Note**

* Prime numbers upto x is approximately: **x/ln(x)** ([hint](https://primes.utm.edu/howmany.html))
* Number of divisors upto n is assumed to be: **∛n** ([hint](https://codeforces.com/blog/entry/14463?#comment-194522))

**Code Template**

|  |  |  |
| --- | --- | --- |
| **Algo Name/Tutorials** | **Code Link** | **Tutorial Link** |
| BigMod, Modular Inverse | [BigMod](https://github.com/bi11a1/Competitive-Programming/blob/master/MyAlgo/bigmod.cpp) |  |
| Bridge, Articulation Point | [Bridge\_CF](https://github.com/bi11a1/Competitive-Programming/blob/master/MyAlgo/bridge_from_CF.cpp), [Bridge\_mine](https://github.com/bi11a1/Competitive-Programming/blob/master/MyAlgo/bridge_mine.cpp) | [Bridge](https://codeforces.com/blog/entry/68138) |
| Prefix-function, KMP | [Prefix-function](https://github.com/bi11a1/Competitive-Programming/blob/master/MyAlgo/prefix-function) | [Prefix-function](https://cp-algorithms.com/string/prefix-function.html) |
| Euclid, Extended Euclidean, Linear Diophantine Equation | [Algebra-Fundamentals](https://github.com/bi11a1/Competitive-Programming/blob/master/MyAlgo/Algebra-Fundamentals.cpp) | [Algebra-Fundamentals](https://cp-algorithms.com/) |
| Bitmask |  | [CF/Errichto-Blog](https://codeforces.com/blog/entry/73558) or [find\_next\_bitset](https://codeforces.com/blog/entry/43718) |
| Finding Cycle | [Cycle](https://github.com/bi11a1/Competitive-Programming/blob/master/MyAlgo/cycle-in-directed-graph.cpp) |  |
| Policy based data structure (pbds) set | [cf](https://ideone.com/QuiYER) | [ordered\_set](https://codeforces.com/blog/entry/11080) or [geeks\_for\_geeks](https://www.geeksforgeeks.org/ordered-set-gnu-c-pbds/) |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Codeforces**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Problem Number** | **Problem Description** | **Solution Hint** | **Important info** |
| 15/03/20 | 1325D-Ehab the Xorcist | Given u and v, find minimum length array such that there XOR=u and SUM=v | Let x=(v-u)/2;  a+b=(a^b) + 2\*(a&b);  So, a&b=(v-u)/2=x;  if(u&x) then ans=[u, x, x]  else ans=[u^x, x] | a+b=(a^b)+2\*(a&b) |
| 15/03/20 | 1323D-Present | Find XOR of all pairwise sums of integers in an array | For each bit k of answer:  i) First mod with 2k+1 and sort  ii) For each element check if sum is in range [2k, 2k+1-1] or [2k+1+2k, 2k+2-2], use binary search to find pairwise range. |  |
| 15/03/20 | 1312D-Count the Arrays | Choose n elements from m numbers, such that there is exactly one duplicate and has increasing and decreasing segments. | mCn-1\*(n-2)\*2n-3 |  |
| 20/03/20 | 1312E-Array Shrinking | Given an array, shrink the array into minimum length. Shrink if a[i]==a[i+1], remove a[i], update a[i+1]=1+a[i+1]. | DP, O(n3)  Consider each segment is shrinkable to 1 length or not using DP. Then use iterative DP to find the maximum reducing size. |  |
| 28/03/20 | 701F-Break Up | Given a weighted graph (n<=103), remove two edges at minimum cost such that after removing them the source and destination become disconnected. | First find the path from source to dest (at most n-1 edges). Then try removing each edge and check whether there exist any bridges. | [Bridge](https://codeforces.com/blog/entry/68138) |
| 29/03/20 | 1327D-Infinite Path | Given a permutation of length n and color of each position, find Pk such that Pk has infinite sequence. | Observe that each number is in a simple cycle. Then for the cycle check for each divisor of the cycle length. |  |
| 30/03/20 | 1327E-Count the block | Given n, count number of block of length 1, 2, .. , n in 0000, 0001, …, 9999 | 2\*10\*9\*10n-L-1 + 10\*(n-L-1)\*9\*9\*10n-L-2  Here, L=length of block |  |
| 02/04/20 | 1326D-Prefix-Suffix Palindrome | From a string S find T such that T is a palindrome and T=a+b; where a is prefix of S and b is suffix of S. | First take max common from both sides. Then use prefix-function for computing palindrome by taking X#X’ | [Prefix-function](https://cp-algorithms.com/string/prefix-function.html)  Visit this frequently |
| 05/04/20 | 471D-MUH and Cube Walls | Given a wall and position, find out if the position's height can be matched with the wall's height by moving up and down. | Get the cumulative difference of the wall's height and position’s height. Then apply KMP. |  |
| 06/04/20 | 808G-Anthem of Berland | Given a text with ‘?’ position and a pattern, count the maximum frequency of the pattern in the text after replacing ‘?’ with any lowercase letter. | Apply DP for replacing each ‘?’ with letter ‘a’ to ‘z’. Then apply KMP. For each possible character and previous match preprocess count nxt[j][ch]. This requires another DP. |  |
| 06/04/20 | 1243D-0-1 MST | Given a weighted graph of weight 0 or 1, find the MST of the graph. Only the edge of weight 1 is given. | Find the number of 0 connected components. Assume all nodes are in the same component then separate them based on 1 labeled edge. Used set. | Increment iterator before deleting element of set during iteration. |
| 06/04/20 | 1243E-Sum Balance | Given k list of numbers, make the summation of each list equal by changing exactly one element from the list to another list. (k<=15, list size=5000) | Use bitmask DP for choosing one of the list and move it to another list. This will form a continuous move and there will be a cycle. If the list value=average then no need to change. | [Masking](https://cp-algorithms.com/algebra/all-submasks.html) |
| 07/04/20 | 1244E-Minimizing Differences | Given an array, find the differences between maximum and minimum elements. You can increase/decrease any element of the array k(<=109) times. | First sort the array. Then use two pointers: left factor and right factor. Whenever we reduce a number check which factor is smaller. Then take that number to its next number. |  |
| 07/04/20 | 1244F-Chips | Given a ring with color black or white, find out the ring’s state after k(<=109) moves. In one move change color[i] if its current color is not equal to any of its adjacent color. | Push segments containing more than one color to a set. Then perform the operation until the set is empty. If one element is not changed then depending on the move number color is decided. |  |
| 08/04/20 | 1244G-Running in pairs | Given n and k, find two permutations such that summation of maximum in each position of the permutations is maximum possible and less than k. | i) if k<(n\*n+1)/2 then impossible  ii) Assume p1=1,2,...,n and p2=1,2,...,n  iii) Now greedily choose if a value from the right side of p1 can be brought forward. |  |
| 08/04/20 | 1228D-Complete Tripartite | Given a disconnected graph, find out if its vertices can be grouped into 3 portions such that there is no edge of the same group and each group has an edge to every other group. | First using dfs make into 3 groups. Then for each vertex check if it is connected to nodes of every other group. |  |
| 08/04/20 | 1216F-Wi-Fi | Given a binary string place either direct link or router in every position. Router can be placed only in ‘1’ position. Each router has a range of k. Cost of a router or direct link is equal to its position. | Use DP. Keep track of the nearest rightmost router and furthest leftmost router. Then do DP from router n-1 to 0.  Note: DP 0 to n-1 fails. Because cost increases linearly in this case. |  |
| 08/04/20 | 1167E-Range Deleting | Given an array, find the total number of pairs such that deleting the number within its range results in a sorted array. | For each l value use two pointers to find its r value. Use segment tree to remove an element/add an element and then verify. |  |
| 09/04/20 | 1333F-Kate and Imperfection | Given a set S={1, 2, …, n}, find out set of length k (k=2, 3, …, n) such that the maximum of pairwise gcd is minimum in the set. | Increase the size of the set one by one. Take a number that has its biggest divisor (less than n) in ascending order. |  |
| 10/04/20 | 133E-Road to 1600 | Given an n by n chess board and queen and rook. Find a permutation of numbers from 1 to n\*n such that following the number rook requires strictly less jump then queen. | Solve this for 3 by 3 grid. Then for each row/column assign a number like a snake. |  |
| 14/04/20 | 1339E-Perfect Triples | A sequence is constructed by choosing minimum positive numbers whose xor is 0 and not presented in the sequence. Now given n find number at position n. | There will be a pattern of 00, 01, 10, 11. To find the number in this pattern find its position in 4n-1. Then mod it by 3 and divide by 3\*4\*4...upto n then mod by 4. |  |
| 21/04/20 | 1337E-Kaavi and Magic Spell | Given two strings S, T and an empty string A, count number of ways to add characters at the front or back of A from S (sequentially) such that A has T as its prefix | Apply DP. Count the number of ways to make T from S. If T is smaller than S then think like this character is equal. Then answer is summation of DP’s from t.length to s.length. |  |
| 21/04/20 | 1270E-Divide Points | Given a set of points, divide them into groups A, B such that for no pair the distance within the same group is equal to the distance between different group | If the parity of x+y is even and odd for some group then the answer is based on that. Otherwise if all of them are even then divide by (x+y)/2, (y-x)/2. If all of them are odd then add (0, 1) with every coordinate. |  |
| 22/04/20 | 1270G-Subset with Zero Sum | Given n integers such that i-n<=a[i]<=i-1, find subsets of those numbers such that summation of them is zero | The formula can be rewritten as, 1<=i-a[i]<=n, Now for each i we can build a directed graph to i-a[i], which will form a cycle. Now consider that cycle of node as the result since their summation will be 0. |  |
| 23/04/20 | 1270F-Awesome Substring | Given a binary string, find the number of pairs of indices such that the number of 1’s in that segment divides the segment length. | We need, pref[r]-pref[l-1]=(r-l+1)/k, where k is some constant.  i) if K is less than sqrt(n) then manual computation  ii) else pref[r]-pref[l-1] will be less than sqrt(n). So we can count the number of 1’s upto sqrt(n) and then find possible values of k. |  |
| 27/04/20 | 975D-Ghosts | Given the starting coordinates of some points and their moving direction as a vector, find the number of collisions. | At the collision point, x1=x2 and y1=y2;  Solving, x1+T.Vx1=x2+T.Vx2 we get a.Vx1-Vy1=a.Vx2-Vy2. Count this and remove the count of pairs if Vx1, Vy1=Vx2, Vy2 | Used map to count point collision and parallel lines |
| 28/04/20 | 976E-Well Played | Given HP and Damage of n enemies, double HP of any enemy (a time max) or Swap damage with HP (b time max). Find maximum obtainable damage. | Note that all the operations of doubling will be applied to only one enemy. So we can iterate through all possible enemies and use cumulative sum for the remaining calculation after sorting based on the difference of HP and damage. |  |
| 28/04/20 | 958B2-Maximum Control (Medium) | Given a tree, place bombs in the leaf upto 1 to n, such that the number of nodes destroyed is maximum. Nodes will be destroyed if it is on the shortest path in between nodes containing bombs. | First find the diameter of the tree. Then take one side of the node in the diameter as root. Now, compute the height of each node. Leaf node has 0 height. Root has maximum height. Now push these values into a priority queue and remove nodes one by one according to their height and add it to the answer. |  |
| 29/04/20 | 883I-Photo Processing | Given an array of size n, divide it into groups of at least k elements such that the difference between maximum and minimum element of each group is minimum possible. | Use binary search on the difference. To validate a value use two pointers. First assume the 0th position is true. Then for first\_pointer from 1 to n if first\_pointer-1 is valid then take the second\_pointer to at least k distance and if val[second\_pointer] - val[first\_pointer] <= mid then make second\_pointer valid. Return true if nth pointer is valid. | Binary Search + Two pointers |
| 29/04/20 | 883K-Road Widening | Given n road lengths and lawn lengths, reduce lawn and add it to roads such that the difference between consecutive roads are at most 1. | Use recursion. Start from the first road and then check what is the maximum road that can be achieved by the next road. If the road length is too large or too small than the previous then it's impossible. Continue until nth road. |  |
| 30/04/20 | 3D-Least Cost Bracket Sequence | Given a bracket sequence including ‘?’ mark, replace the ‘?’ to make a valid sequence. For each ‘?’ mark there is a cost associated with it to make it either ‘(‘ or ‘)’. Show the minimum cost and final string. | Iterate through the string. Count in opening or decrease otherwise. Replace each ‘?’ mark with ‘)’ and add the cost. In a priority queue keep the difference of cost. So, if at any point if count reaches negative then replace the ‘?’ that has maximum revenue with ‘(‘. |  |
| 30/04/20 | 831C-Jury Marks | Given, some scores by some judge and some middle value of total score after starting from x. Count the total possible values of x. | Find the cumulative sum of the scores given by judges. Then for one known value compute x. x=csum-b[0]. Then validate x for each b. |  |
| 01/05/20 | 287B-Pipeline | Given, some numbers 1, 2, …, k. Make exactly n using minimum total numbers. | If (k\*k+1)/2 >= n then we can always make n. It's optimal to take numbers from largest. Binary search on the starting position. | Can always make n using 1, 2, …, k if (k\*k+1)/2>=n |
| 01/05/20 | 688D-Reminder Games | Given n numbers and k you need to say if x%k can be calculated if x%ni is given | We can compute x%k from x%a if a=ck  Also if x%a is known and x%b is known then x%lcm(a, b) can also be known | Given x%a and x%b, we can calculate x%lcm(a, b) |
| 03/05/20 | 1348E-Phoenix and Berries | Given n shrubs containing blue and red berries, you can either take k blue/red berries in the same basket or take k blue+red berries of the same shrub in one basket. (n<=500) | Use DP. Try to keep t red berries apart from the current shrub. If we know there are t red berries left then we can count the remaining blue berries. DP will say either yes or no. If we can leave t red berries. Note there will be at most one basket from the same shrub containing a mixture of red and blue berries. |  |
| 04/05/20 | 1348F-Phoenix and Memory | Given some range of valid positions of numbers, find out if each number can be placed uniquely. | First sort the segments and then for each position add the segments of left within its position and then take the minimum segment of right value. To find whether a second arrangement exists, check if two numbers can be swapped using priority queue. |  |
| 05/05/20 | 632E-Thief in a Shop | Given n items with its profit, you have to pick exactly k values. How many different sums can you obtain? (n,k<=1000) | First we normalize the given values by subtracting the minimum value. So, we will have 0 in it. Then we try to make 1 to n\*k using minimum items. For this we use DP. First set DP to infinite. Then for each element j,  DPi=min(DPi, DPi-ara[j])  In the final answer we check if DPi<=k then we add i+k\*mn to the answer. |  |
| 06/05/20 | 817F-MEX Queries | Given n queries of [r, l] <= 1e18 you have to either set or reset or inverse values in range. Then find MEX of the array. | Used bitset. First take all unique ranges into points. This way size will be within range 2e5. Take total bitset of value 2e5 with group of 32 or 64. Then for each query apply operation to a group or iterate through first and last group. For query check whether all bits of a group are set or not. Note for bitset complexity reduces by a factor of 32 or 64. | Bitset |
| 05/05/20 | 664C-International Olympiad | You are given the starting year as 1989. This year will be represented by 9, then year 1990 by 0, 1991 by 1, and so on. If an abbreviation is assigned take one more digit. Now given the short form find the year. | Note that the first 10 years take 1 digit. Next 100 years take 2 digits. Next 1000 years take 3 digits and so on. Now we know if the short form is k digit then in which segment of year it will fall. Now start with the short form and add the latest ten power until it falls within that segment. |  |
| 08/05/20 | 1345E-Quantifier Question | Given a relation of the for (x1<x2) && (x3<x4) && … (xi<xj) assign universal/existential operator to each variable such that the statement is true and number of universal statements is maximized. | First built a directed graph from xi->xj.  i) If exists cycle then no answer  ii) For each node check if there exist a smaller index then this node in the forward and reverse order using dp  iii) If no smaller value found then assign it as universal else existential | [cycle-in-directed-graph](https://github.com/bi11a1/Competitive-Programming/blob/master/MyAlgo/cycle-in-directed-graph.cpp) |
| 10/05/20 | 1345F-Resume Review | Given an array a[], choose values b[i] from 0 to a[i], such that sum(b[i]\*(a[i]-b[i]2)) is maximized and sum(b[i])=k; n<=105, k<=109 | If increase value by 1 then,  increase=x(a-x2)-(x-1)(a-(x-1)2)  =a-3x2+3x-1  Now we can greedily choose k times which value gives us the maximum. But since k is large we can see that this increase decreases gradually. So, binary search on that value and check if we can make it decrease upto that value using k operations. |  |
| 16/05/20 | 472D-Design Tutorial: Inverse the problem | Given a matrix of distances of a tree find out if the distance matrix is valid for a tree with positive edge weight. | Use MST to build the tree then check if the distance matrix of that tree matches with the given matrix. Note that the distance between two nodes is strictly less than the distance between further nodes. So, MST will choose direct edges first. |  |
| 18/05/20 | 1354C1/C2-Polygon Embedding | Given 2\*n vertices of convex polygon find minimum enclosing square. | First build the polygon. Find its diagonal. Then use binary search to rotate it and calculate minimum x and y coordinate ranges. |  |
| 19/05/20 | 1354D-Multiset | You are given a multiset of size n (<=10^6). You have to add some element from it. Or remove the k'th element. Print any element of the resultant multiset. (Memory limit: 28MB) | Note that we could have used a pair of set for this if the memory limit was bigger. Keep a global cnt, at each iteration increment it and add {value, cnt} to the set. To remove an element just find upper\_bound {value, 0} and remove it.  -To solve it in the given memory limit use segment tree. At each node keep count of values below that node. To remove an element go either left or right depending on the required position of k. | Multiset using segment tree. |
| 19/05/20 | 1354E-Graph Coloring | Given a graph (multiple components), find if it can be colored with 1, 2, 3 such that the difference of color for each adjacent node is 1. Also the number of nodes colored with 1, 2, 3 is n1, n2 and n3 consecutively. | Note that each edge of the resultant graph will be either 1-2 or 2-3. That means the graph needs to be bipartite. We check how many nodes with color 2 is required if we color one component and vice versa. Then apply DP to find the suitable combination of the color. If we can not find such a combination then the answer is no. Otherwise we color the nodes according to DP. |  |
| 20/05/20 | 1354F-Summoning Minions | Given n minions with their strength a, b and k limits. You can add 1 minion and all active minions strength will increase by b. There can not be more than k minions at a time. You can remove some minions. | Use DP.  First note that, we should remove minions only after adding k-1 minions. Then by removing minions we are increasing the value of k-1 minions. Note that we can not remove after k because it will already have k limit and can not add a new one to remove. Now to choose k-1 minions we need to sort the array based on b. So that if we choose from beginning to end it will always have greater value of b and will get added with more minions. | Did it myself :D |
| 20/05/20 | 1354G-Find a Gift | Interactive problem. You are given n items either rock or gift. Weight of rock is greater than the weight of gift and equal. But the weight of gifts might vary. In each query you give two lists of indexes and the query will tell which group is heavier. Find the minimum index that contains a gift. | First, we check if index 1 can be a gift. For that we use 30 random checks with index 1. If its weight is smaller then its a gift. Then it has a (1-2-30) chance of being a rock. Then we check range [1] and [2], [1, 2] and [3, 4], [1,2,3,4] and [5,6,7,8] …. . If any portion is not equal then it contains a gift. Then we binary search on that portion to find out the gift’s position. To know the gift’s position we compare it with the previous rock segment. | Interactive |
| 22/05/20 | 1355E-Restorer Distance | Given heights of n items built with bricks, make all of them equal height by adding/removing/moving some bricks. Each operation has some cost associated with it. | Note that we can apply ternary search on the height. And if we move away from optimal results the value will keep increasing. So, we can ternary search on height and calculate optimal results. | For ternary search,  m1=low+(high-low)/3  m2=high-(high-low)/3 |
| 22/05/20 | 1355A-Sequence with Digits | You are given a1 and you have to find ak. Where, k<=1016 an+1=an+max\_digit(an)\*min\_digit(an) | Note that, after some operation some digit will become 0. So, mx\*mn will be 0 and the value will remain the same for the rest of the operations. |  |
| 22/05/20 | 1355C-Count Triangles | Given a, b, c, d (<=5\*105), find number of valid triangles in range a<=x<=b<=y<=c<=z<=d | Note that it's optimal to only check if z<x+y or not. Because x<y+z and y<x+z will automatically be true for the range condition.  Soln-1: we calculate the answer for each y in range [b, c]. For each y, find the possible value of x=max(a, c-z+1). Then we find out the number of ways to increment z and then apply summation of series formula.  Soln-2: we calculate number ways to form numbers greater than z by using x+y. For each x in range [a,b] we use prefix-sum in range [a+b]++, [a+c+1]--. Then we use cumulative sum to find number of ways greater than [x+y]. Then foreach z we add ways[x+y+1] to the answer. | For arithmetic series,  Sn=n/2\*{2a+(n-1)\*d}  an=a+(n-1)\*d |
| 23/05/20 | 1355F-Guess Divisor Count | Interactive: There is a value X (<=109)  In each query you will give Q (<=1018) and get the value of gcd(X, Q). Find the number of divisors in X such that the relative error is not more than 0.5 and absolute error is not more than 7. | We will try to find the common divisors with X for each prime upto 1000. We will multiply the answer by 2 or add 7. Calculate the exact answer for smaller primes. Still the number of queries will be large for primes upto 1000. We will have to check for prime upto 630. After knowing prime factors to calculate exact prime factors we will take two factors raised to its maximum power and then ask query. | Interactive |
| 27/05/20 | 1358E-Are you Fired? | Given n numbers where first (n+1)/2 are given and rest are x. Find an appropriate value of k such that the summation of each array of length k is positive. | Observe that if there is an answer with length n/2, we can multiply it by 2 and still it will be a valid answer. So, for each length greater than n/2, we check whether values with this length are positive. To find out we use cumulative sum of differences and take its maximum. That means we check how much the value is incremented and how much it decrements. Similar to a two pointer approach. |  |
| 01/06/20 | 1363F-Rotating Substring | Given two strings S, T, perform minimum number of operations such that S=T. In one operation you can rotate counterclockwise any substring of S once. | Use DP.  Note that, rotating a substring once is equivalent to taking a character from any position and putting it into any position before its current position.  So, we start from end of S and T. Then check if S[p1]==T[p2], then we can call DP[p1-1][p2-1]  If they are not equal then we can remove the last character from S[p1] and put it before anywhere else. Which will be counted as 1 operation. We go to state DP[p1-1][p2]  We can also check if number of characters upto p1 in S is less than number of characters upto p2 in T. Which means we chosen some character of S and place it in here to obtain state DP[p1][p2-1] |  |
| 02/06/20 | 1311D-Three Integers | You will be given three integers x, y, z. You have to convert them into a, b, c by performing minimum number of operations such that a|b and b|c. In each operation you can either increase or decrease x, y, z also they must remain positive. (x,y,z<=104) | Value of b should not be greater than 2\*104. Otherwise we could simply take 2\*104 as b, c. We iterate through all possible values of b and find all of its divisors and choose the minimum cost one. To choose c we can divide c by b and floor and ceil the value to choose the nearest divisor of c. | Harmonic series:  n+n/2+n/3+...+n/n  n(1+½+⅓+...+1/n)  =nlogn |
| 02/06/20 | 1311E-Construct the Binary Tree | Construct a binary tree containing n nodes and summation of depth of each node should be equal to d. | Try to build a complete binary tree with n nodes. If their summation of depth is greater than d then the answer does not exist. Now keep the leftmost leaf separately and for each leaf to its right try to add it below this leaf if possible. Otherwise go parent of this leaf until it can be added. If there is no child left but still d is not reached then the answer is also no. Otherwise show the tree. |  |
| 03/06/20 | 1311F-Moving Points | Given n points on OX axis along with their velocity. Find summation of minimum possible distance for all pairs of points. Two points can intersect at any real positive time t. | Note that two points will intersect only if p1.x<p2.x && p1.v>p2.v. Otherwise the initial distance is the minimum distance among them.  First we sort the points decreasingly based on their velocity. Then for each point i, find out the number of points which are smaller than the point i. Add them to the anwer. Use segment tree for finding points less than current point and use cumulative sum on each valid segment. |  |
| 06/06/20 | 1362E-Johnny and Grandmaster | You will be given n numbers in the form pk. Where n values of k will be given. Make two groups such that the difference between them is minimum. | We will try to reduce the difference between two groups. First sort the numbers based on their power. Then iterate from largest to smallest power. If two powers are the same then assign them to different groups. Otherwise if one is pk and another is pk-x, then we need to find px numbers of pk-x. Because px \* pk-x=pk. We keep searching until powers are matched or there are no elements left. |  |
| 07/06/20 | 1362F-Johnny and Megan’s Necklace | You are give n pairs of necklace part. You have to merge them into one cyclic necklace. Maximum joining cost is considered as the cost of building. Find the maximum cost. Cost of join is xor of joining parts that divides 2k. | We can merge two necklace part if the last k bits are same for both of them. For each necklace pair form an edge in a graph with a[i]&(2^k-1) and b[i]&(2^k-1). Then we try to find eulerian path of that graph. If there exist the path then we build the cycle. | Eulerian Cycle |
| 08/06/20 | 1365F-Swaps Again | You will be given two arrays a, b. You have to determine if a can be converted into b after performing some operations. In each operation you can swap prefix of length k with suffix of length k. Here k must be <=n/2 | Note that after performing one operation the pair a[i], a[n-i-1] remains the same. So, we make all possible pairs of array a and b. And check if both of them are equal or not. If equal then it's possible to construct b from a. Otherwise not possible. Note that we can swap positions of a[i], a[n-i+1] by making a move of length i and then i-1. Also if pairs of both arrays are equal then we can construct the array b starting from middle position. |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**SPOJ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Problem Number** | **Problem Description** | **Solution Hint** | **Important info** |
| 12/04/20 | LSORT-Sorting is not easy | Given a permutation of 1 to n, sort them by removing one element one by one and appending it into the left or right of a new list. Cost of each operation at i’th iteration=i\*pos[ara[i]] | Start from the sorted list. Precalculate position of an element if you delete range from left to right. Then apply DP to erase elements from either left or right. |  |
| 28/04/20 | PARTIT-Partition | Given n (<=10), m (<=220), k (<=220), find the lexicographically kth array such that a[1]+a[2]+...+a[n]=m and in non-descending order. (Test Case <=100000) | Use DP[first\_number][m][n], where first\_number is the starting combination of the array and need to make m using n elements and DP stores the total number of combinations. In the query, assume i th number is cur(start with 1). If DP is < K then subtract that value and increase cur. Repeat n times. |  |
|  |  |  |  |  |
|  |  |  |  |  |

**CP-Algorithm**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Problem Number** | **Problem Description** | **Solution Hint** | **Important info** |
| 15/04/20 | UVA-11029 Leading and Trailing | Find the leading 3 and trailing 3 digits in nk. (1<=n<=232, 1<=k<=109) | i) Trailing digits can be obtained by (nk)%1000  ii) To obtain leading 3 digits: Let, nk=r  klogn=logr  r=10klogn  p=klogn  (10p-floor(p))\*100 | [Problem](https://vjudge.net/contest/368283#problem/C) |
| 16/04/20 | SPOJ-LOCKER | Given length n(<1012), divide n into some integers such that their multiplication is maximum possible. | i) If 3 | n express n as summation of 3’s  ii) If after modding n by 3 ther result is 1 then add 4\*summation of 3’s in (n-4)  iii)else 2\*summation of 3’s in (n-2) | [Problem](https://vjudge.net/contest/368283#problem/F) |
|  |  |  |  |  |
|  |  |  |  |  |