2020 “Lenovo Cup” USST Campus Contest Solutions

University of Shanghai for Science and Technology

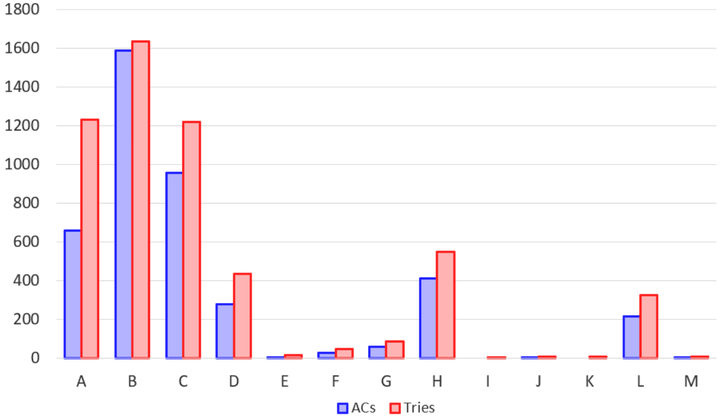
May 30, 2020

Expected Difficulty

- Very Easy: ABC
- Easy: DHL
- Medium Easy: GEJM
- Medium Hard: FIK
- Hard: ?

Problems and # of Participants Solved

A	B	C	D	E	F	G	H	I	J	K	L	M
661	1589	958	279	6	28	59	412	0	2	0	215	3



B. Bamboo Leaf Rhapsody

- First solved: Xu BenQi 0:03 (+)
- Brief: You are given n points in 3D space, find the nearst point from $(0, 0, 0)$.
- The answer is $\min\limits_{1 \le i \le n} \{\sqrt{{x_i}^2 + {y_i}^2 + {z_i}^2}\}$.

A. Archmage

- First solved: Gu YiZhen 0:04 (+)
- Brief: Archmage has n mana at most and can cost x mana to cast once per second and restore y mana once per second. How many times he can cast at most in m seconds?

Observation

- $x + y \leq n$
- If $x \leq y$, he can cast every second.
- If $x > y$, all mana in the first $m - 1$ seconds will not be wasted.

- So the answer is $\min(\lfloor \frac{n+(m-1)y}{x} \rfloor, m)$.

C. Cheat Sheet

- First solved: Wu Ke 0:04 (+)
- Brief: You are given a paper which can write n characters at most and m words to write. The words written on the paper must be separated by 1 space. How many distinct words can be written on the paper?

- There may be duplicate words, so we can use std::set or something else to keep diffierent words.
- After that, we can write the shorter words greedily.

L. Lottery Tickets

- First solved: Zhang QingChuan 0:23 (+1)
- Brief: You have n numbers from 0 to 9, and there are c_i number i. Find the largest integer can be constructed which is a multiple of 4.

- Enumerate the last two digits of the answer.
- When the last two digits are determined, the largest integer is determined too.
- There may be some corner cases: single $0, 4$ or 8, leading zeros...

H. Hay Mower

- First solved: HexHexHex 0:34 (+2)
- Brief: $n \times m$ matrix. Number in grid (i, j) will be added $a_{i,j}$ every second. k operations. Each operaation will add one column or one row to the sum, then trun them to 0. Find the sum in the end.

Observation

- The contribution of one grid is only determined by the last operation changed it.
- We can easily maintain the last time of the operation about row i or column j, denoted as r_i and c_j.
- So the last change of the gird (i, j) is at $\max(r_i, c_j)$.

- We can get the sum in $O(n \times m + k)$.

D. Disaster Recovery

- First solved: Liu MingJie 0:22 (+1)
- Brief: You are given an undirected connected graph with n vertexes and m edges. The cost of an edge connected i and j is $\mathrm{Fib}_i + \mathrm{Fib}_j$. Find an MST which the largest degree of the vertex in the MST is minimized.

Observation

- The cost of an edge may be very large.
- The sum of any two diffierent terms of Fibonacci sequence is not equal.
- A well-known property: the costs of edges in the graph are different from each other, so the MST is unique.
- The size relation of an edge (x, y) is equivalent to the size relation of pair$(\max(x, y), \min(x, y))$.

- First solved: Lan TianLang 3:37 (+2)
- Brief: You are given an $n \times m$ 01 matrix. You are required to infer the matrix in 2000 queries at most. (Check the statement for more details)

Observation

- Consider a 2×3 matrix $\begin{bmatrix} A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,1} & A_{2,2} & A_{2,3} \end{bmatrix}$.
- We can infer if $A_{1,2}$ euqals to $A_{2,2}$ by asking the whole 2×3 martix and two 2×2 submatrixes.

J. JXC!!

- Ask each 2×2 submartrix. ($(n-1)(m-1)$ queries)
- Each 2×3(or 3×2) submartrix can get the information: two elements are equal or not. ($nm - 5$ queries)
- Use one query 2 to ask any element not on the corner, then we can determine the whole matrix, except for the corner. (1 query)
- Use four queries 2 to ask the elements on the corner. (4 queries)
- Output the answer. (1 query)
- $(n-1)(m-1) + nm + 1$ queries is enough to determine the whole matrix. For $n = m = 32$, it equals to 1986.

E. Eight Digital Games

- First solved: Gu YiZhen 1:25 (+2)
- Brief: There is a string consists of only numbers from 1 to 8. Each kind of inversion has its own cost. You can also change all x into y, y into x at a specific cost. Minimize the total cost.

- All operations are actually mapping one permutation to another.
- There are few kinds of inversion. We can get the number of each kind of pair in $O(8n)$, so we can quickly count the cost of inversion after mapping.
- Consider each operation as an edge, each permutation as a vertex. We can get the cost of mapping with Dijkstra Algorithm.

M. MITE

- First solved: Ji XingLong 3:37 (+)
- Brief: Plant sugar cane on $n \times m$ matrix. Sand block can be replaced with water block, but stone block cannot. Sugar cane can be planted on the grid adjacent to at least one water block. Construct a scheme that can plant the most sugar cane.

Observation

- m is small. This suggests that we can use dynamic programming.
- After the decision of line i, the number of sugar cane that can be planted depends on the status of line $i-1$ and line $i-2$.
- The state of the last two lines can be abstracted into a ternary number.

G. Gentle Jena

- First solved: Dong Shi 2:08 (+9)
- Brief: Append numbers to the end of the sequence dynamically, calculate the sum of RMQ of all subintervals in the sequence every time. Your solution should be online.
- The range of n is very large. So we need a linear algorithm.
- Cartesian tree can help us to reduce the difficulty of thinking, but if we only know the monotone stack, we can also solve it.

Observation

- Consider the number of times x in a position appears as the minimum value of the interval. Once a number larger than x is appended, the contribution of x will not change.
- When a number is appended, only the contribution of the number monotonically increasing from the end to the front will change.
- Use a monotone stack to maintain the numbers(It's actually the right most chain of Cartesian trees).

K. K-Shift Array

- Brief: There is an array of length n and m operations. Each operation (l, r) will K-Shift the subinterval $[l, r]$ or ask the sum of subinterval $[l, r]$. (Check the statement for more details)

- Consider the solution when $k = 2$.
- The K-Shift operation is equivalent to the swap of the numbers with odd index and the numbers with even index in the subinterval.
- Use two Splays(or Treaps) to maintain the odd index numbers and the even index numbers respectively.
- The K-Shift operation is actually the swap of two corresponding subtrees of two Splays.
- When k is not fixed, we need $\mathrm{lcm}(1, \cdots, k)$ Splays(or Treaps).

F. Fake Algorithm

- First solved: Lv YiQiang 0:36 (+)
- Brief: Divide integers into minimum number of groups so that integers in each group are coprime with each other. You should hack the greedy strategy by constructing the input.

Observation

- The original problem can be easily reduced to the minimum coloring of graphs.
- Let each number correspond to a vertex. If two numbers are not coprime with each other, add an edge between them.
- The minimum coloring of the graph is the answer.

F. Fake Algorithm

- Because the numbers can be repeated, the problem is much easier than our expection.
- Repeat the sample k times is enough to pass.

- Bonus:
- Try to find the input of distinct numbers.
- Try to find the minimum size of the input for each k.

I. Immortal Trees

- Brief: Calculate the number of spanning trees of n vertexes. There are some limitations: some vertexes have limitations of upper or lower degrees, and some edges must exist.

Observation

- It's easy to think of the Prufer code.
- But some edges are fixed, so it is difficult to use the Prufer code directly.
- We might as well reduce each connected component to one vertex.

- Assuming that there are m distinct edges and they not form a circle, then n vertexes are divided into $n - m$ connected component, equivalent to a $n - m - 2$ long Prufer code.
- Try to use DP to solve it.

- Let $F_{i,j}$ be the number of plans that the first i connected components have used j positions in $n-m-2$ long Prufer code.
- It is easy to get

$$F_{i,j+k} = \sum_{k=0}^{n-m-2-j} \binom{n-m-2-j}{k} \times F_{i-1,j} \times g(i,k+1)$$

- where $g(i,k+1)$ represents the number of plans that the i-th connected component has connected $k+1$ edges to other connected components(regardless of fixed edges in this connected component).
- We find that $g(i,k+1)$ is the same problem, so we can use another DP to calculate it.

I. Immortal Trees

- Let $G_{i,j,k,l}$ be the number of plans that the i-th connected component connects j edges to other connected components,where the first k vectexes has used l edges.
- Let the size of the i-th connected component is sz_i, where the j-th vertex is $p_{i,j}$.
- Let L_i, R_i respectively represent the lower degree and the upper degree of the i-th vertex(after removing fixed edges).
- It is easy to get

$$G_{i,j,k,l+o} = \sum_{o=L_{p_{i,k}}}^{\min(R_{p_{i,k}},j-l)} \binom{j-l}{o} \times G_{i,j,k-1,l}$$

- We have $g(i, k+1) = G_{i,k+1,sz_i,k+1}$.
- In a way, it's like a Knapsack problem.

I. Immortal Trees

- Some corner cases:
- There are duplicate limitations.
- If the fixed edges happen to construct a complete tree, special judgment is needed.
- If there are circle in fixed edges, the answer must be 0.

Thank you for your participation.