2017 KAIST RUN Spring Contest with HYEA Cup Solution (English)

HYEA (ho94949, hyea@kaist.ac.kr)

Sponsored By:

Startlink

Naver D^2

1 Card Game Contest

Consider multiplication principle. If there are a ways of doing something and b ways of doing another thing, there are total $a \times b$ ways performing both actions. Answer will be product of $max(A_i, 1)$.

You should carefully handle the case for $M = 1$, especially when $N = 0$. $N = 0$ and $M = 1$ should give 0 for the answer since every integer modulo 1 is 0.

Trivia: The case $N = 0, M = 1$ is intended. Be careful when you calculates modulo with $M = 1$.

2 Mahjong

If you have less than 4 tiles in your initial hand, the tile has probability for waiting. For each added tile, check whether it is winning hand or not.

You have to check 7 Eyes as exception. Now, let's work on 4 Melds and 1 Eye case.

One way for checking winning hand is consider every combination of $\binom{14}{2\ 3\ 3\ 3\ 3}$, which gives 538137600 ways, for each 9 waitings, leads TLE. You can improve this by changing same tile does not affect combination and using which number head should be, since sum of every meld is divided by 3.

Another way and also model solution is backtracking for all possible 4 Melds. You can set your Eye and using there can only be 16 types of meld, you can backtrack through this. This will give $9 \times 9 \times 16^4 = 5308416$ calculation ways. You can reduce it anyway.

Maybe you can solve this using calculating waiting directly, using that there are 5 types of waiting. (waiting Eye, waiting Meld of same type, waiting Straight in middle, waiting Straight both side, waiting Straight in one side)

Trivia: I first wanted to give calculate value of 14 hands with given rule, but then thought this is not IPSC (2016 M - Mana troubles, which is advantageous knowing Magic The Gathering.)

3 The Way

Notice that going leftward only can happen in middle line. Think going S or Z-shaped path passing middle line as one partition of way and use Dynamic Programming.

More precisely, every path can be divided by the part S or Z-shaped part and its partition is unique. So you can make dynamic table by $D_{i,j}$ as ends with (i, j)-grid and not going back further.

Trivia: First, $N = 10^6$ but we reduced the problem. This can be solved using matrix and fast exponential algorithm giving time complexity $O(logN)$

4 The Other Way

Calculate shortest path of every city using Dijkstra's algorithm. Now, roads used in shortest path forms Directed Acyclic Graph. so you can run Dynamic Programming here.

You don't have to explicitly make the graph but calculate only the difference of two nodes are length of the road.

Trivia: SPFA will not solve this. I put anti-SPFA data. Thanks for alex9801 making data for this.

5 Just as Tic Tac Toe

First, you can calculate Grundy number of each basket, by calculating remainder divided by $M+1$. Grundy number has important property that If you play multiple games, It's Grundy number will be bitwise exclusive or of games. Now problem changes to 3XOR problem. It's same with calculating a xor $b = c$, thus we should find way calculating a xor b fast. For each n bits, calculate Fourier transform for convolution gives n-dimensional Fourier transform. It is called Hadamard transform will solve this problem.

Fast Walsh-Hadamard Transform matrix looks like this:

$$H_N = \frac{1}{\sqrt{2}} \begin{pmatrix} H_{N-1} & H_{N-1} \\ H_{N-1} & -H_{N-1} \end{pmatrix}$$

Trivia: I first wanted to give this problem as Set game with multiple characteristics. Then, you should use matrix really looks like FFT matrix. I reduced this for kind of normal case.

6 Balance

Let the answer of this problem as $A(N)$. In the last step of placing N weights, you can put any of $2^1, \cdots, 2^{N-1}$ weights to both side, and you only can put 2^N to right side.

It gives $A(N) = (2N-1)A(N-1)$. So the answer is $1 \times 3 \times \cdots \times (2N-1)$.

Trivia: There are $O(N^2)$ DP solution. Well optimized-solution failed to solve this problem with method directly. I opened various possibility for solving this problem by setting $N = 50000$.

7 Memory

Minimum possible action is $RC/2$ since every card should be flipped to be removed. Consider following flipping action. First, flipping two cards has shape A, B. Second, flipping two cards has shape C, A. Third, remove both A. And card that once flipped has shape A, C. It is one of the worst case. In this case, we can work on this recursively, giving $RC-1$, since removing two cards needs two step. This bound is optimal since one can flip $RC-2$ cards in $RC/2-1$ steps, flip one of last two cards make you know every information of cards, making you to do additional $RC/2$ flip to remove all cards.

Trivia: You can solve this by dynamic programming using array $D_{i,j}$ as calculating maximum number of possible step, when i cards exists and j cards were shown.

8 Too Many Traps

The answer for this question is always positive, but when blanket is placed where treasure is. Constructive proof can be given and it is solution for this problem. (number of blanket should be less than number of treasure. In this proof, $M = N-1$)

Let $A_1 < A_2 < \cdots < A_N$ and use induction. If $N = 0, 1$, problem is trivial.

1. A_N is one position of blanket and none of $A_1, \cdots, A_{N-1}$ are:

   You can jump starting from A_N and solve problem for remaining $N-1$ jumps. swap first two jumps, gives you also does not jump on blanket of position A_N

2. A_N is not one position of blanket and one of $A_1, \cdots, A_{N-1}$ are:

   You can jump starting from A_N and solve problem for remaining $N-1$ jumps. You escaped one blanket before A_N, therefore you don't have to consider.

3. A_N is one position of blanket and one of $A_1, \cdots, A_{N-1}$ are:

   for pair $(A_1, A_1 + A_N), \cdots, (A_{N-1}, A_{N-1} + A_N)$; by Pigeonhole principle, at least one of the pair meets both number is not the distance blanket placed. So, jumping A_i, A_N gives escaping two blankets and you can solve problem for remaining $N - 2$ jumps.

4. neither of $A_1, \cdots, A_N$ are not position of blanket.

   You can solve problem starting from A_N and jumps with $A_1, \cdots, A_{N-1}$ without considering nearest blanket. If you jumps on nearest blanket, swap A_N and next jump of nearest blanket A_i. It will make every jump before A_N lands before nearest blanket. and further jumps after A_N do not jumps on the other blanket.

So you can implement this carefully. The most time consuming-part will be third one which can be $O(NlogN)$ but amortized complexity will give you $O(logN)$.

Implementing this will give you complexity $O(N^2)$.

You have to improve two part; whether one of $A_2, \cdots, A_N$ are position of blanket. And, for the fourth case, check whether there exists position lands to nearest blanket.

First one can be improved by, checking only first some of position of blanket. The important part is not exactly jumping into blanket but skipping them. Second one can be improved by, Binary-indexed tree-like structures, subset sum with update (you have the swap part, so dynamic update is needed.) And, binary search for jumps lands to first blanket.

Implementing this will give $O(Nlog^2N)$, which is model solution. You can improve it by $O(NlogN)$ solving dynamic subset sum problem and binary search problem together.

Trivia: The first solution was $O(N^2)$ and zlzmsrhak gave idea for reducing it to $O(NlogN)$. One main reason of giving this problem as $N = 100000$ is to prevent random shuffle algorithm with some tricks like maximum jump is crucial solving this problem. This problem is the hardest one in the contest. You can really be proud of you if you solved this problem!

9 Protocol

$112345^{167} \equiv 1 (\text{mod } 1000000009)$.

So you only should have to consider about modulo 167 of capacity. Use matrix multiplication and fast-exponential algorithm for solving this.

More precisely, you can give 167×167 matrix with $A_{(ax^2+bx+c,x)} = 1$ and $A_{(dx^2+ex+f,x)} = 1$ (If two coincides, value is 2), other is 0. A^M will give you how wires change after M years.

Trivia: This is one main reason why I prepared this contest. I just want let you know that constant 112345 is awesome!

10 Communism

Halve your array and calculate $\pm(A_1 \text{ or } 0) \pm (A_2 \text{ or } 0) \pm \cdots$. You can sort two array in similar sense with merge sort. put every element and sorting with complexity $O(N \times 3^{(N/2)})$ will give you TLE. If one element of array is chosen, calculate possible interval of another array. You can use sliding window for this technique. Total time complexity is $O(3^{(N/2)})$.

Trivia: My fast possible implementation of $O(N \times 3^{(N/2)})$ was 1.2 seconds. Some normal and implementation using C++ Standard Library was 0.6 seconds. You may thought this problem does not work on $O(3^{(N/2)})$ but computers were fast!

11 Thanks

Thank you everyone who worked for contest.

Thank you for Maisie, Licia, alex9801, 64bitfox, Narae, Kir, ReUnite, moraeduji, HYEA, Kahr, Rinkaru, Adnim and LemmonBerry who allowed me to use their name in description.

Thank you for myself, who prepared this problem set, worked on problems.

Thank you for alex9801, for solving problems, editing description and advice for problem Mahjong, The Other Way, Balance and Memory.

Thank you for wwiiiii, leader of RUN@KAIST contacting sponsor and let me hold this contest with my name.

Thank you for koosaga, for solving problems, giving lots of off-by-one solutions or other wrong answered greedy-like solution, advice for problem Communism.

Thank you for functionx, for solving problems, giving greedy-like solution for problem Too many traps, advice for overall problem difficulty.

Thank you for zlzmsrhak, for solving problems, giving core idea of improvement of problem Too Many Traps, advice for problem Card Game Contest, Just as Tic Tac Toe, Balance, Too Many Traps and overall problem difficulty.

Thank you for Naver D^2 for sponsoring Money for running contest and prize.

Thank you for baekjoon and Startlink let us using online judge service acmicpc.net.

And finally, thank you for everyone who participated in contest and watching this solution.