Problem Tutorial: "Aftermath"

Let $d_1 < d_2 < \ldots < d_k$ be the divisors of n, let $\sigma_0(n)$ be the number of divisors of n, and $\sigma_1(n)$ be the sum of divisors of n.

First, $a = \frac{d_1+d_2+\ldots+d_k}{k} = \frac{\sigma_1(n)}{\sigma_0(n)}$.

Second, $h = \left(\frac{d_1^{-1}+d_2^{-1}+\ldots+d_k^{-1}}{k}\right)^{-1} = \frac{nk}{n(d_1^{-1}+d_2^{-1}+\ldots+d_k^{-1})}$.

At this point, notice that $\frac{n}{d_i} = d_{k-i+1}$.

Thus, $h = \frac{nk}{d_k+d_{k-1}+\ldots+d_1} = \frac{n\sigma_0(n)}{\sigma_1(n)}$.

Then, $a \cdot h = \frac{\sigma_1(n)}{\sigma_0(n)} \cdot \frac{n\sigma_0(n)}{\sigma_1(n)} = n$.

Problem Tutorial: "Believer"

Let's solve the problem subject to $\sum_{i=1}^{k} a_i \le n$. It's easy to prove this is equivalent to $\sum_{i=1}^{k} a_i = n$.

Consider some integer x. If we don't include x into a, then $c(f(a,x)) = 0$. To make $c(f(a,x)) = p$, we need to include x into a exactly $2^p - 1$ times. It means we have to include x into a once to increase the answer by 1, include it 2 more times to increase the answer again, include it 4 more times to increase the answer again, and so on.

Thus, for every positive integer x, consider a set of integers $S(x) = \{x, 2x, 4x, 8x, ...\}$. Then, consider the union of these sets as a multiset. From this multiset, we have to pick as many elements as possible with sum at most n.

Clearly, it's optimal to pick integers in non-decreasing order. Thus, we'll be able to pick all occurrences of integers up to some t and several occurrences of $t+1$.

Let's use binary search on the largest picked integer t. If we can find the sum of all occurrences of integers up to t, the problem is solved.

Let $\sigma(t) = \frac{t(t+1)}{2}$. The smallest elements of $S(x)$ are $1, 2, 3 \ldots$, and the sum of those which don't exceed t is $\sigma(t)$. Similarly, the i-th smallest elements of $S(x)$ are $2^{i-1}, 2 \cdot 2^{i-1}, 3 \cdot 2^{i-1} \ldots$, and the sum of those which don't exceed t is $2^{i-1} \cdot \sigma(\lfloor \frac{t}{2^{i-1}} \rfloor)$. Since $\sigma(0) = 0$, we only need to consider $i \le \log(t) + 1$. Thus, the sum can be calculated in $\mathcal{O}(\log t)$.

The overall complexity of the solution is $\mathcal{O}(\log^2 n)$.

Problem Tutorial: "Chalk Outline"

Pictures tell it better...

For $k < n - 3$, the answer is "`No`", since any polygon can be triangulated, which leads to existence of at least $n - 3$ internal diagonals.

For $k \ge n - 3$, the answer is "`Yes`".

When $k = n - 3$, draw $n - 1$ points (i, i^2) and one point $(0, -10^9)$.

For larger k, pick the first t points and "reflect" them so that they form a convex polyline and all diagonals between them are internal, but all diagonals with the rest of the points (except the $(0, -10^9)$ one) are not internal. This way, we'll get $\frac{t(t-3)}{2} + 1$ extra diagonals.

To get more diagonals, slowly move the $t+1$-th point down. It will be able to "see" the first t points one by one from left to right.

Thus, to get exactly k diagonals, pick the largest t such that $n - 3 + \frac{t(t-3)}{2} + 1 \le k$. By "reflecting" the first t points and stopping the movement of the $t+1$-th point at the right moment, we'll solve the problem.

Problem Tutorial: "Do I Wanna Know?"

Let $f(n, k)$ be the probability that in a tournament for n players, there are k players who defeated each of the remaining $n - k$ players.

Let's calculate $f(n + 1, k)$. Consider the $n + 1$-th player. If she belongs to the losing group, she loses to some k other players which have ID less than hers, so the probability of this event is p^k, and then the probability that there are k winning players among the first n is $f(n, k)$. She may also belong to the winning group with probability $(1 - p)^{n-k+1}$, and then we are left with $k - 1$ winning players among the first n, which happens with probability $f(n, k - 1)$. Thus:

$$f(n + 1, k) = f(n, k) \cdot p^k + f(n, k - 1) \cdot (1 - p)^{n-k+1}.$$

Let's calculate $f(n + 1, k)$ again, but considering the first player instead of the $n + 1$-th player this time. Analogously, we'll arrive at:

$$f(n + 1, k) = f(n, k) \cdot (1 - p)^k + f(n, k - 1) \cdot p^{n-k+1}.$$

Now we see that:

$$f(n, k) \cdot p^k + f(n, k - 1) \cdot (1 - p)^{n-k+1} = f(n, k) \cdot (1 - p)^k + f(n, k - 1) \cdot p^{n-k+1};$$
$$f(n, k) \cdot (p^k - (1 - p)^k) = f(n, k - 1) \cdot (p^{n-k+1} - (1 - p)^{n-k+1}).$$

We know that $f(n, 0) = 1$, and then we can calculate $f(n, k)$ from $f(n, k - 1)$ by the formula above for all k from 1 to $n - 1$.

A special case is $p = \frac{1}{2}$: the formula above becomes $f(n, k) \cdot 0 = f(n, k - 1) \cdot 0$, which is not very helpful. But due to symmetry of $p = 1 - p$, it's easy to find the answer: $f(n, k) = \binom{n}{k} \cdot \frac{1}{2^{k(n-k)}}$.

Problem Tutorial: "Exit Song"

We are given two generators $r_i = f(r_{i-1})$ and $s_i = g(s_{i-1})$. These generators have preperiod and period of length at most $n + m$.

Let's ignore preperiod (it can be handled easily). Let p_f be the period of f and p_g be the period of g. Without loss of generality, assume p_f and p_g to be coprime. (Otherwise, we just solve the same problem $gcd(p_f, p_g)$ times.)

Let $h(x) = g(g(...g(x)...)) = g^{p_f}(x)$. Then, for each row r, for some $x(r)$, the occupied seats will be $x(r)$, $h(x(r))$, $h(h(x(r)))$, and so on. The number of these seats is $\lfloor \frac{k}{p_f} \rfloor$ or $\lfloor \frac{k}{p_f} \rfloor + 1$.

Take all occupied seats in some row and add all of them to a set — some data structure which allows adding seats, removing seats and calculating the number of segments of consecutive empty seats (it's easy to do every operation in $\mathcal{O}(\log m)$). Then we know the answer for this row.

Then, consider some other row r' such that either $x(r) = x(r')$ or $x(r') = h(x(r))$. If there is no such row, let r' be an imaginary row with $x(r') = h(x(r))$ and the same number of occupied seats as in r. For row r', the set of occupied seats differs from the same set for row r in $\mathcal{O}(1)$ elements.

If we process the rows in right order, the total number of changes to the set will be $\mathcal{O}(n + m)$, since we'll do $\mathcal{O}(1)$ changes per row and per seat number. The overall complexity of the solution is $\mathcal{O}((n+m) \log m)$.

Problem Tutorial: "Forever and Always"

Create $\frac{p(p+1)}{2} + 2$ voters and $p + 1$ options.

For every option i from 1 to p, let there be one voter with list $\langle i, p + 1 \rangle$ and $i - 1$ voters with list $\langle i \rangle$. Let there also be two voters with list $\langle p + 1 \rangle$.

In the first iteration, the number of votes per option will be $(1, 2, \ldots, p, 2)$.

In the second iteration, the person who voted for option 1 and has option $p+1$ in their list will change their vote to option $p+1$. The distribution of votes will become $(0, 2, 3, \ldots, p, 3)$.

In the third iteration, the person who voted for option 2 and has option $p+1$ in their list will change their vote to option $p+1$. The distribution of votes will become $(0, 1, 3, 4, \ldots, p, 4)$.

The remaining iterations will happen similarly, and the final distribution of votes after the $p+1$-th iteration will be $(0, 1, 2, 3, \ldots, p-2, p-1, p+2)$.

Problem Tutorial: "Gate 21"

Let a_i be the y-coordinate of the visited checkpoint at $x = i$, and let $d = a_2 - a_1$ be the "slope" of the line. Then, $a_i = a_1 + d \cdot (i-1)$. Since $l_i \le a_i \le r_i$, we have $l_i \le a_1 + d \cdot (i-1) \le r_i$. We are asked to find the number of pairs (d, a_1) such that the inequality is satisfied for all i.

The inequalities are nothing else but semiplane inequalities on a plane with coordinates d and a_1. The answer is the number of integer points inside the intersection of $2n$ semiplanes. This intersection can be found in $\mathcal{O}(n)$ since semiplanes are already sorted.

Counting integer points inside a polygon with potentially non-integer vertex coordinates might look tricky, but in fact, all semiplanes intersect any line $x = d$ at some integer y. Thus, as we move a pointer to d from left to right, we'll be able to calculate the answer as the sum of several arithmetic progressions defined by two semiplane inequalities.

Problem Tutorial: "Hamilton"

Since we can just reverse the path and nothing changes, let's assume $a < b$.

The answer is "`No`" if n is even while both a and b are odd: a valid path requires two even integers to stand next to each other, which is not allowed.

If $a = 1$ and $b = n$, the answer is 0 and the path is $\langle 1, 2, \ldots n \rangle$.

If $b - a = 1$, the answer is 1 and the path is $\langle a, a-1, \ldots, 1, n, n-1, \ldots, b \rangle$.

Otherwise, the answer is almost always 2, sometimes 1, and sometimes 3. After careful consideration of a handful of cases, it's possible to get this problem accepted.

The problem can be solved by a clever brute force, pretty much avoiding any cases. Let's call a cut a connection between points i and $i+1$ which is not walked by. The number of cuts is equal to the number of flights, since every time we don't walk between consecutive cells, we have to fly.

Clearly, since we start at a, there is a cut either between $a-1$ and a or between a and $a+1$. We can check both options. Similar argument applies to b. Now we have several segments of cells. We can try all permutations of segments and the direction in which every segment is walked by and check if an answer is possible.

Unfortunately, sometimes making cuts adjacent to a and b is not enough. The smallest case when the answer is 3 is $n = 15$, $a = 11$, $b = 13$.

It turns out that making an additional cut between cells 1 and 2 is always enough in this case. Strictly proving this fact might still need some case analysis, though.

It was also possible to get this problem accepted by trying a random additional cut several times — there are plenty of suitable additional cuts for large cases.

Problem Tutorial: "I've Got Friends"

The problem asks to restore multigraph G by its line graph $L(G)$.

First, let's get rid of multiedges. If two vertices in $L(G)$ are connected and have the same set of neighbors, then if there exists a solution, a solution where these edges are equal in G exists as well. Thus, we can remove one of these vertices in $L(G)$.

Wikipedia page on line graphs is helpful to get the general idea. Whitney's isomorphism theorem states that for any two graphs G and G' with at least 5 vertices there is a bijection between isomorphisms of G and G' and isomorphisms of $L(G)$ and $L(G')$. This means that if we have restored G from $L(G)$ and it contains at least 5 vertices, then G is unique up to isomorphism.

The solution for each connected component goes as follows:

- Add vertices of $L(G)$ into consideration one by one in some traversal order. We need every newly added vertex to have at least one neighbor in $L(G)$ already added.
- While there are less than 5 vertices in G, find any G with brute force.
- When there are at least 5 vertices in G, we need to check if a new edge can be added to G which is incident to a given set of edges of G (its neighbors in $L(G)$). If these edges are all incident to one vertex in G, the new edge can be created connecting this vertex and a new vertex. If these edges can be covered by two vertices in G which are not connected by an edge, the new edge can be created connecting these two vertices. Otherwise, the answer is "`No`".

The complexity of this solution is $\mathcal{O}(n+m)$.

Problem Tutorial: "Joke"

The second player can refuse to cover any cards and take everything the first player can lay down. The first player will therefore lose once she's left with jokers only.

This strategy doesn't work for the second player if she has both jokers initially. In this case, the first player can lay down cards until the second player finally covers something. The second player will be able to get rid of at most one joker this way. Then the first player wins by taking cards from the second player until she's left with the second joker.

Problem Tutorial: "Kids Aren't Alright"

Let $n = p_1^{a_1} p_2^{a_2} \dots p_k^{a_k}$. First, we need to find $a_1, a_2, \dots, a_k$. Since $n \le 10^{18}$, we can first try to factorize it dividing by all integers up to 10^6. Then the remaining part is either a prime p, a square of a prime p^2, or a product of two primes pq. The first case can be detected using Miller-Rabin test or `java.math.BigInteger.isProbablePrime()`. The second case can be detected by taking the square root. If the first two cases don't take place, the third one does.

The set with GCD = 1 and LCM = n must consist solely of divisors of n. The converse doesn't hold: GCD and LCM might turn out to be divisors of n, not necessarily 1 and n.

Now, use inclusion-exclusion principle. For every prime divisor p_i of n, GCD is either divisible by p_i or not. Similarly, LCM is either divisible by $p_i^{a_i}$ or not. This is defined by whether there exists an integer in the set which is not divisible by p_i, and whether there exists an integer in the set which is divisible by $p_i^{a_i}$, correspondingly.

Thus, there are four options for each prime divisor of n. After deciding that, it's easy to calculate t, the number of integers which might be in the set, as the product of a_i+1, a_i or a_i-1 for all i. Then we must add $2^t - 1$ to the answer with an appropriate sign.

This is a bit too slow, since the complexity of this solution is $\mathcal{O}(4^k)$ for $k \le 15$ (this is the maximum number of prime divisors of an integer $n \le 10^{18}$).

There are various ways to optimize this solution. The most natural way is to notice that two of the four described options (for every prime divisor) lead to equivalent computations, so instead of branching four times, we can branch just three times. This way, the complexity is $\mathcal{O}(3^k)$ which is enough.