UBC Summer Contest 2018

Contents

A: Andrew and Efficient Change 2

B: Badminton's Paul 3

C: Cyclic Song 4

D: David vs David 5

E: Enegue's Enigmatic Lanterns 6

F: Forever Young 7

G: Jonathan and Jason at the Jowling Jalley I 8

Q: Quirky Queries 9

A: Andrew and Efficient Change

Author: Gregory Zhang

First, we calculate the minimum number of coins needed to make each value $x \leq r$. We can compute this in $O(nr)$ using a DP. Let the set of current coin values be C. Then, our DP is

$$\text{dp}[x] := \text{ minimum number of coins to make x,}$$
$$\text{dp}[x] = \begin{cases} 1 + \min_{v \in C} \text{dp}[x-v] & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ \infty & \text{if } x < 0 \end{cases}.$$

Now, suppose we have some value x, and we want to find the coin value to add that minimizes the number of coins we need to use to make x. Suppose we want to add a coin with value v. Then, we can iterate over each multiple of v that is at most x, and keep track of the minimum number of coins. That is, we compute

$$\text{minimum coins after adding } v = \min_{0 \leq k \leq x/v} k + \text{dp}[x - kv].$$

Observe that doing this calculation for each $v \leq x$ takes $O(x \log(x))$ because we get a harmonic sum. Then, since $r - l \leq 50$, we simply try adding each $v \leq r$, and get the minimum number of coins after adding v in $O((r-l)r \log(r))$.

Complexity: $O(nr + (r-l)r \log(r))$.

B: Badminton's Paul

Author: Henry Xia

Suppose that the tree is a path. Then, consider the 2D plot with time on the x-axis, and the edges on the y-axis. Label the edges as $e_1, e_2, ..., e_{n-1}$ in order. A delivery from a to b using the edges $e_l, ..., e_r$ during the time s to t would be represented as the lattice points $\{(x, y) : s \leq x < t+1 \text{ and } l \leq y < r+1\}$. To get the cost for all deliveries, we find the area of the union of the rectangles. We can answer each query in constant time by storing the area strictly left of c_j and $d_j + 1$ for each query from time c_j to d_j. This can be done in $O((m+q)\log(m+q))$ by sweeping along the x-coordinates while maintaining a segment tree on the compressed y-coordinates.

To solve this problem on a tree, we consider the heavy-light decomposition of the tree. Then, each delivery will intersect $O(\log(n))$ chains in the decomposition. If we put the edges on the y-axis such that each chain is a contiguous segment, we will get $O(\log(n))$ rectangles for each delivery. We can proceed as in the path case with rectangle union.

Complexity: $O((m\log(n)+q)\log(m\log(n)+q))$.

C: Cyclic Song

Author: David Berard

Observe that the description of a valid song of type N is actually describing a binary de Bruijn sequence of order N. It is known that the shortest de Bruijn sequences of order N have length 2^N. Furthermore, we know that de Bruijn sequences of order N can be represented as Eulerian circuit in the $(n-1)$-dimensional de Bruijn graph. Now, this problem becomes finding an Eulerian circuit in the $(n-1)$-dimensional de Bruijn graph of 2 symbols that minimizes the distance from the edge representing S to the edge representing T.

Suppose we have some trail containing S and T, then we can complete the Eulerian circuit in linear time if the resulting graph is connected. Now, it suffices to show that the distance from S to T is never too big, so we can try all short enough trails that start with S and end with T.

Conjecture 1. Let A denote the node representing $\underbrace{AA...A}_{N-1 \text{ A's}}$, and let B denote the node representing $\underbrace{BB...B}_{N-1 \text{ B's}}$.

Case 1: If the tail of the edge representing S is the same as the head of the edge representing T, and this node is either A or B, then the minimum distance from S to T is either $2^N - 1$ or $2^N - 2$.

Case 2: Otherwise, the minimum distance from S to T is at most $N + 2$.

Proof. **Proof of Case 1**. Without loss of generality, suppose the node is A. Since we need an Eulerian circuit, and we start at A, we need to visit all the other edges before coming back to A or we get stuck at A.

Proof of Case 2. The graph is very nice (diameter N, 2-connected if we stay away from A and B, etc.). Chuck believes it. □

In **Case 1**, we have a trail from T to S, and we can complete that Eulerian circuit in linear time. In **Case 2**, we have up to $N + 2^1 + 2^2$ trails from S to T, each of which we can check in linear time.

Complexity: $O(N2^N)$.

D: David vs David

Author: Henry Xia

Looking at small initial states of the game suggests the following lemma.

Lemma 1. Let (x_i, y_i) be the i-th losing position, with $i \geq 0$ and $x_0 = y_0 = 0$. Let $S_n = \{x_i : i \leq n\} \cup \{y_i : i \leq n\}$ be the numbers that appear in losing positions up to n. Then, $x_i = \text{mex}(S_{i-1})$ and $y_i = x_i + (k+1)i$, where $\text{mex}(S)$ is defined to be the smallest non-negative integer that is not an element of S.

Proof. It suffices to check that we cannot reach a losing position from another losing position, and that we can always reach a losing position from a winning position. □

Now, this problem gets rude. Observe that we will eventually see every positive integer in a losing position, and no positive integers will appear in two different losing positions. Furthermore, observe that $\{x_n\}_n$ and $\{y_n\}_n$ are roughly linear. Hence, we check that $\{x_n\}_n$ and $\{y_n\}_n$ are complementary Beatty sequences. Suppose that $x_n = \lfloor \alpha n \rfloor$ and $y_n = \lfloor \beta n \rfloor$ where $1/\alpha + 1/\beta = 1$. Now, since $x_n + (k+1)n = y_n$, we get $\lfloor \alpha n \rfloor + (k+1)n = \lfloor \beta n \rfloor$, so we can set $\beta = \alpha + k + 1$. Solving for α and β gives irrational numbers for all $k \geq 0$, so we do have complementary Beatty sequences. Now, we can binary search to check whether the positions given in the input appear at a losing position.

Complexity: $O(n \log(\max(x, y)))$.

E: Enegue's Enigmatic Lanterns

Author: Rehim Memmedli

Let $f(n)$ denote the number of composite divisors of n. Observe that for no integer $a \in [4, 100]$ do we have $f(a) = f(a-1) = f(a-2) = f(a-3)$. Hence, to solve the problem for n lanterns of which k are on, we only need to check all subsets of size $n - m$ where m is the smallest positive integer such that $f(k) \neq f(k-m)$. If the response to the query is not $f(k)$, then we know that the lanterns that we excluded from the subset must be on. Since we know that $m \leq 3$, we need to check at most $\binom{n}{3} \leq 2 \times 10^5$ subsets.

Complexity: $O(n^4)$.

F: Forever Young

Author: Andrew Lin

Longest Increasing and Decreasing Subsequences.

The number of permutations with a longest increasing subsequence of length n and a longest decreasing subsequence of length m is the number of standard Young tableaux with n rows and m columns. Since $s-(n+m) \leq 50$, we can brute force over all possible Young diagrams with s cells, n rows, and m columns (there are as many Young diagrams as ways to partition $s-(n+m)$). Then, we can use the hook-length formula to count the number of standard Young tableaux for each Young diagram. Hence the name "Forever Young".

Complexity: $O(k^2 p(k))$, where $k = s-(n+m)$ and $p(k)$ is the number of partitions of k.

G: Jonathan and Jason at the Jowling Jalley I

Author: Eugene Shen

Observe that there is a bijection between valid positions of n rows and Dyck paths from $(0,0)$ to $(2n+2,0)$. Hence, the number of valid positions of n rows is C_{n+1}, where C_{n+1} is the $n+1$-st Catalan number. We can easily hardcode the first 21 Catalan numbers.

Complexity: $O(1)$.

Q: Quirky Queries

Author: Brandon Zhang

Observe that if a and b are square-free, then the sorted sequence of divisors of a is lexicographically smaller than the sorted sequence of divisors of b if and only if the sorted sequence of prime divisors of a is lexicographically smaller than the sorted sequence of prime divisors of b. Since the described quirky integers are square-free, we only need to consider prime divisors.

Now, define a new order on the integers which is the lexicographic order of their sorted prime divisors. Hence, type 1 queries are "set $a_i = \min(a_i, x)$ for all $i \in [l, r]$". Also, type 2 queries are simply set union. Assuming that we can do set union in $O(1)$, these two queries can be done in amortized $O(\log^2(n))$ by using a segment tree that stores the two largest elements and the set union of each segment. Since there are only 62 prime numbers less than 300, we can store the prime divisors of each number using 64-bit integers as bitmasks. This allows constant time lexicographical comparison and set union (left as an exercise).

Complexity: $O(q(\log^2(n) + 62))$.