

Problem F

Beautiful Currency

原案：岩田
解答作成：平澤，北川
解説作成：北川

問題概要

- 数列 a_1, a_2, \dots, a_n が与えられる
- $a_i \rightarrow b_i$ に変更して $b_i \mid b_{(i+1)}$ を満たすようにしたい
- 変更するコストは $|a_i - b_i| / a_i$
- コストの最大値を最小化

解法

- $b_{(i+1)}$ の条件は b_i にしか依存しない → DP
- $dp[i][j] = (\text{a}_1 \sim \text{a}_i \text{ の部分問題の } \text{a}_i = j \text{ になる場合の最適解})$
- $dp[i+1][j] = \min_k \{ \max(dp[i][k], \text{a}_{(i+1)} \rightarrow j \text{ のコスト}) \}$
- k は j の約数を動く

高速化

- j の約数を求めるのは遅いので、逆に k を固定してその倍数を更新していく

```
for(int i = 0; i + 1 < n; ++i) {  
    for(int k = 1; k < 上限; ++k) {  
        for(int l = 1; k * l < 上限; ++l) {  
            dp[i+1][k*l] の更新  
        }  
    }  
}
```

上限の見積もり

- $2^k \leq a_i \leq 2^{(k+1)}$ のとき $a_i \rightarrow 2^k$ にすれば、最悪コスト 0.5 で可能
- よって $b_n \leq 1.5 * a_n$

計算量

- 上限を N とすると
- $O(n \cdot (N/1 + N/2 + \dots + N/N)) = O(n \cdot N \log N)$
- $n \leq 20$ 、 $N \leq 1.5 \cdot 10^5$ なので間に合う

```
for(int i = 0; i + 1 < n; ++i) {  
    for(int k = 1; k < 上限; ++k) {  
        for(int l = 1; k * l < 上限; ++l) {  
            dp[i+1][k*l] の更新  
        }  
    }  
}
```

Result

- First Accept: usaaaaagi (12min)