

水以下コン (チーム戦) 解説

運営メンバーのなまえ

2024/03/30

A: Cyan or Less

Writer: Kyo_s_s

A: Cyan or Less

カラーコードが文字列で与えられる.

彩度を以下で定義する:

- r, g, b をカラーコードの RGB 値とする.

$$\text{彩度} = \frac{\max(r, g, b) - \min(r, g, b)}{\max(r, g, b)}$$

与えられたカラーコードが表す色は水色 #00c0c0 の彩度以下か判定せよ.

A: Cyan or Less 解法

step1. 与えられたカラーコードを RGB 値に変換する.

- 2桁の16進数を10進数に変換するには, 1つ目の数字を16倍して2つ目の数字を足せばよい.
 - a は 10 に, b は 11 に, ..., f は 15 に変換してから足す.

A: Cyan or Less 解法

step1. 与えられたカラーコードを RGB 値に変換する.

- 2桁の16進数を10進数に変換するには, 1つ目の数字を16倍して2つ目の数字を足せばよい.
 - a は 10 に, b は 11 に, ..., f は 15 に変換してから足す.

step2. 彩度を計算する.

- 彩度の定義通りに計算すれば OK.

A: Cyan or Less 解法

step1. 与えられたカラーコードを RGB 値に変換する.

- 2桁の16進数を10進数に変換するには, 1つ目の数字を16倍して2つ目の数字を足せばよい.
 - a は 10 に, b は 11 に, ..., f は 15 に変換してから足す.

step2. 彩度を計算する.

- 彩度の定義通りに計算すれば OK.

step3. 彩度が水色の彩度以下か判定する.

- 水色 #00c0c0 も同じ方法で彩度を計算しておけば, 大小関係を比較するだけ.

A: Cyan or Less 余談

実は、水色 #00c0c0 の彩度は 1

→ 彩度がこれより大きくなる色は存在しない！

このため、すべてのケースで Yes と正解すればよいです。

一旦没になったのですが、丁寧に書いてもよいし、気づけば1行で解けるので面白くない？となり出題されました。

X: $f(f(f(f(f(x))))))$

Writer: Kyo_s_s

X: $f(f(f(f(f(x))))))$

整数 K ($1 \leq K \leq 10^{18}$) と x についての関数 $f(x)$ が与えられる． 最初 $x = 1$ として，次の操作を K 回繰り返す：

- x を $f(x)$ で更新する

最終的な $x \bmod 998$ は？

X: $f(f(f(f(f(x))))))$

整数 K ($1 \leq K \leq 10^{18}$) と x についての関数 $f(x)$ が与えられる． 最初 $x = 1$ として，次の操作を K 回繰り返す：

- x を $f(x)$ で更新する

最終的な $x \bmod 998$ は？

構文解析をがんばる必要がある ... ？

→ めちゃめちゃな式は与えられないため，そんなに頑張らなくてよい．

X: $f(f(f(f(f(x))))))$ 部分点1 解法

- $K = 1$, $f(x)$ に $*$, \wedge は含まれない.

X: $f(f(f(f(f(x))))))$ 部分点1 解法

- $K = 1$, $f(x)$ に $*$, $^$ は含まれない.

$f(x)$ は, x もしくは 1 以上 10^9 未満の整数 の和で表されているので,

step 1. $f(x)$ を $+$ で分割する

step 2. 分割したそれぞれの文字列を数値に変換する (x なら 1)

step 3. すべて足し合わせる

を実装すれば OK.

X: $f(f(f(f(f(x))))))$ 部分点 2 解法

- $K \leq 10^4$, $f(x)$ に \wedge は含まれない.

X: $f(f(f(f(f(x))))))$ 部分点 2 解法

- $K \leq 10^4$, $f(x)$ に \wedge は含まれない.

部分点 1 解法で, 「数値 / x と演算子 $+$ のみからなる数式」を計算した.
→ 改造すると, 「数値 / x と 演算子 $*$ のみからなる数式」を計算できる.

X: $f(f(f(f(f(x))))))$ 部分点 2 解法

- $K \leq 10^4$, $f(x)$ に \wedge は含まれない.

部分点 1 解法で, 「数値 / x と演算子 $+$ のみからなる数式」を計算した.

→ 改造すると, 「数値 / x と 演算子 $*$ のみからなる数式」を計算できる.

step 1. $f(x)$ を $+$ で分割する

step 2. 分割したそれぞれの文字列は「数値 / x と 演算子 $*$ のみからなる数式」なので, それぞれを計算する

step 3. すべて足し合わせる

そのままだとオーバーフローするため, 和 / 積の計算時に $\text{mod } 998$ を取れば OK.

X: $f(f(f(f(f(x))))))$ 部分点 3 解法

- $K \leq 10^4$

X: $f(f(f(f(f(x))))))$ 部分点 3 解法

- $K \leq 10^4$

「数値 / x と 演算子 $*$, $^$ のみからなる数式」を計算できれば OK. これは,

step 1. $f(x)$ を $*$ で分割する

step 2. 分割したそれぞれの文字列は「数値 / x もしくは, 演算子 $^$ のみからなる数式」なので,

- $^$ を含む: (数値 or x) $^$ (数値) の形になっているので計算する
- $^$ を含まない: そのまま数値に変換する

step 3. すべて掛け合わせる

とすれば実装できる. 愚直に K 回シミュレーションすれば OK.

X: $f(f(f(f(f(x))))))$ 満点解法

X: f(f(f(f(f(x)))))) 満点解法

関数 $f(x)$ の形から, x を更新するときに $x \bmod 998$ で更新してもよい. つまり, 関数 f は,

$$f : \mathbb{Z}/998\mathbb{Z} \rightarrow \mathbb{Z}/998\mathbb{Z}$$

とみなせる. \rightarrow ダブリングできる!

X: $f(f(f(f(f(x))))))$ 満点解法

関数 $f(x)$ の形から, x を更新するときに $x \bmod 998$ で更新してもよい. つまり, 関数 f は,

$$f : \mathbb{Z}/998\mathbb{Z} \rightarrow \mathbb{Z}/998\mathbb{Z}$$

とみなせる. \rightarrow ダブリングできる!

前処理として $x = 0, 1, \dots, 997$ に対する $f(x)$ を計算しておくことにより, $x = 0, 1, \dots, 997$ に対する $f(f(x))$ の値が計算でき, さらにこの値を用いて x に対する $f(f(f(f(x))))$ の値が計算でき, ...

これを繰り返し, 必要な部分を適用することで K 回操作した後の値を求めることができる!

X: $f(f(f(f(x))))$ 余談

- 原案では 引き算 - と 括弧 () も含まれていたのですが、さすがにやりすぎということで無くしました。
 - 括弧が入ってくるとちゃんと再帰的な処理をする必要があります。
- ^ もなくていいじゃん、と言われていたのですが、^ が無いと Python の eval を使うことで構文解析をサボれてしまうのでやむなく入れました。
 - 部分点 2 までは eval をやるだけで通せます。
 - 部分点 3 も、`re.sub(r'(x|\d+)\^\d+', r'pow(\1,\2,998)', S)` と置換することで、構文解析パートは eval で済ませることができます (ダブリングはする必要があります)。

X: Unions

Writer: yasunori

X: Unions

N 個の国と、複数の国で構成される同盟が M 個ある. i 個目の同盟には C_i 個の国 $A_{i,1}, A_{i,2}, \dots, A_{i,C_i}$ の国が所属している.

同じ同盟に所属している国同士は直接行き来でき、同盟 i に属している国同士は D_i 分で移動できる. 同じ同盟に属していない国同士は直接行き来できない.

国 $2, 3, \dots, N$ について、国 1 から移動するのにかかる時間の最小値を求めよ.

- $2 \leq N \leq 10^5$
- $1 \leq M \leq 10^5$
- $\sum_{i=1}^M C_i \leq 10^5$

X: Unions 部分点 1 解法

- $C_i = 2 \ (1 \leq i \leq M)$

X: Unions 部分点1 解法

- $C_i = 2 \ (1 \leq i \leq M)$

すべての同盟が2個の国で構成されているため、それぞれの同盟を「 D_i 分かかる道でつながっている」と言い換えることができる。

X: Unions 部分点1 解法

- $C_i = 2 \ (1 \leq i \leq M)$

すべての同盟が2個の国で構成されているため、それぞれの同盟を「 D_i 分かかる道でつながっている」と言い換えることができる。

つまり、国1から国 $2, 3, \dots, N$ への最短距離を求める問題に帰着できる。

→ これはダイクストラ法で解ける！

国1を始点として、ダイクストラ法を使って各国への最短距離を求めることで、部分点1に正解できる。

X: Unions 部分点 2 解法

- $\sum_{i=1}^M C_i \leq 10^3$

X: Unions 部分点 2 解法

- $\sum_{i=1}^M C_i \leq 10^3$

ある同盟 i について、その中の国同士は直接行き来でき、移動にかかる時間は D_i である.

つまり、同じ同盟に属する国同士は「 D_i 分かかる道でつながっている」と言い換えることができる.

X: Unions 部分点 2 解法

- $\sum_{i=1}^M C_i \leq 10^3$

ある同盟 i について、その中の国同士は直接行き来でき、移動にかかる時間は D_i である.

つまり、同じ同盟に属する国同士は「 D_i 分かかる道でつながっている」と言い換えることができる.

$\sum_{i=1}^M C_i \leq 10^3$ の制約から、各同盟について同じ同盟に属する国同士のペアすべてに D_i の重みを持つ辺を張っても十分間に合う.

部分点 1 と同様にダイクストラ法を用いれば、部分点 2 に正解できる.

X: Unions 満点解法

X: Unions 満点解法

部分点 2 の方法だと，張る辺の本数が多すぎて間に合わない．

超頂点を導入することで間に合う！

X: Unions 満点解法

部分点2の方法だと、張る辺の本数が多すぎて間に合わない。

超頂点を導入することで間に合う！

各国 $1, 2, \dots, N$ と、同盟 $1, 2, \dots, M$ を頂点とするグラフを考える。

$i = 1, 2, \dots, M, j = 1, 2, \dots, C_i$ について、

- 国 $A_{i,j}$ から 同盟 i に、重み D_i の辺を張る
- 同盟 i から 国 $A_{i,j}$ に、重み 0 の辺を張る

とすると、同じ同盟に属する国同士は同盟の頂点を経由することで D_i 分のコストで移動できる。

このグラフの辺数は $2 \times \sum_{i=1}^M C_i$ であるため、このグラフ上でダイクストラ法をすればよい。

X: X-word Database

Writer: Kyo_s_s

X: X-word Database

整数 X ($4 \leq X \leq 10^5$) と文字列 S ($1 \leq |S| \leq X$) が与えられる.

以下の条件を満たす文字列を **よい文字列** と呼ぶ:

- 文字列の長さが X 以下
- 辞書順で S 以下
- 連続部分列に cyan を含む

よい文字列は何個ある？

X: X-word Database

整数 X ($4 \leq X \leq 10^5$) と文字列 S ($1 \leq |S| \leq X$) が与えられる.

以下の条件を満たす文字列を **よい文字列** と呼ぶ:

- 文字列の長さが X 以下
- 辞書順で S 以下
- 連続部分列に cyan を含む

よい文字列は何個ある？

$X = 5$, $S = \text{cyan}$ のとき, 条件を満たす文字列は,

acyan, bcyan, ccyan, cyan, cyana, cyanb, cyanc

の 7 個.

X: X-word Database 部分点 1 解法

- $X \leq 8$

X: X-word Database 部分点 1 解法

- $X \leq 8$

cyan を含む長さ 8 以下の文字列を全て試せば OK.

たとえば長さが 8 で cyan を含む文字列は,

****cyan, ***cyan*, **cyan**, *cyan***, cyan****

のどれかの形なので, * に入るアルファベットを全て試してそれぞれがよい文字列かどうかを判定すればよい!

X: X-word Database 部分点 1 解法

- $X \leq 8$

cyan を含む長さ 8 以下の文字列を全て試せば OK.

たとえば長さが 8 で cyan を含む文字列は,

****cyan, ***cyan*, **cyan**, *cyan***, cyan****

のどれかの形なので, * に入るアルファベットを全て試してそれぞれがよい文字列かどうかを判定すればよい!

このままだと $X = 8$ で cyancyan を 2 回数えてしまうのでそこだけ注意.

X: X-word Database 部分点 2 解法

- $X \leq 10^3$, S はすべて z で長さが X の文字列

X: X-word Database 部分点 2 解法

- $X \leq 10^3$, S はすべて z で長さが X の文字列

長さが X 以下で cyan を含むような文字列はすべてよい文字列になる.

X: X-word Database 部分点 2 解法

- $X \leq 10^3$, S はすべて z で長さが X の文字列

長さが X 以下で cyan を含むような文字列はすべてよい文字列になる.

長さが K で cyan を含む文字列の個数は, 包除原理を使うと,

$$\sum_{i=1}^{\lfloor K/4 \rfloor} (-1)^{i+1} \cdot 26^{K-4i} \cdot {}_{i+1}H_{K-4i}$$

で求められる (ここで, ${}_nH_k$ は重複組合せ).

X: X-word Database 部分点 2 解法

- $X \leq 10^3$, S はすべて z で長さが X の文字列

長さが X 以下で cyan を含むような文字列はすべてよい文字列になる。
長さが K で cyan を含む文字列の個数は、包除原理を使うと、

$$\sum_{i=1}^{\lfloor K/4 \rfloor} (-1)^{i+1} \cdot 26^{K-4i} \cdot {}_{i+1}H_{K-4i}$$

で求められる (ここで, ${}_nH_k$ は重複組合せ). よって, 答えは,

$$\sum_{K=4}^X \sum_{i=1}^{\lfloor K/4 \rfloor} (-1)^{i+1} \cdot 26^{K-4i} \cdot {}_{i+1}H_{K-4i}$$

X: X-word Database 満点解法

X: X-word Database 満点解法

桁 DP で解ける！

$\text{dp}[i][\textit{smaller}][j] := X$ の上から i 番目まで見て,

$\textit{smaller} = 0$: X と同じ / 1 : X より小さく,

cyan の j 文字目までを末尾に含む / すでに cyan を含む
文字列の個数

X: X-word Database 満点解法

桁 DP で解ける！

$\text{dp}[i][\text{smaller}][j] := X$ の上から i 番目まで見て、

$\text{smaller} = 0$: X と同じ / 1 : X より小さく、

cyan の j 文字目までを末尾に含む / すでに cyan を含む
文字列の個数

上から DP をしていくことで、 X 字以下の文字列の個数も求められている。

$$\sum_{i=0}^X (\text{dp}[i][0][\text{cyan を含む}] + \text{dp}[i][1][\text{cyan を含む}])$$

が答え。