

H: Noise Cancelling

原案:climpet

問題文:riantkb

データセット:climpet

解答:climpet, hos

解説:climpet

問題概要

- $(1, 2, -3, 4, -1, -2, 3, -4)$ のように、前半の各要素を -1 倍すると後半に等しくなる偶数長の数列を、noise-cancelling 列 (NC 列) と呼ぶことにする。
- 長さ n の数列 A が与えられる。 A に連続部分列として含まれる NC 列はいくつあるか。出現位置の異なるものは区別する。

制約

- $n \leq 2 \times 10^5$

記号などの定義

- この解説では、以下の記号等を使用します。
 - $S[l..r]$ 数列 S の、 l 番目から r 番目の要素からなる連続部分列
 - $S[l..]$ 数列 S の、 l 番目以降の要素からなる接尾辞
 - $\text{lcp}(S, T)$ 数列 S と T の最長共通接頭辞長
 - $\text{neg}(S)$ 数列 S の各要素を -1 倍した数列
- 解法は少なくとも 3 種類あります。

解法1 (by climpet)

- ある数列 S について、 $y[i] = \text{lcp}(\text{neg}(S), S[i..])$ という数列 y を考える。
 - この配列自体は、Z アルゴリズムを用いて線形時間で計算可能。
- run の列挙アルゴリズムの 1 つである Main–Lorentz 法は、分割統治と Z アルゴリズムを用いて実装できる。ここで、Z アルゴリズムを上記の y で置き換えると、 $A[i] = -A[i + \text{周期}]$ という定義上での run (のようなもの) を列挙していることに相当する。
- 全体で $O(n \log n)$ 時間。

解法2 (by potato167)

- 各正整数 k について
 - 長さが $2^{k+1} \times (\text{奇数})$ と書ける NC 列のみを考える。
 - 数列 C_k を、 $C_k[i] = (i \text{ の下位 } k \text{ ビット目が } 0 \text{ の場合 } A[i], \text{ そうでない場合 } -A[i])$ として定義する。
 - A 上の NC 列は、 C_k 上では単なる square (同じ列の 2 回の繰り返し) として出現する。
 - よって、 C_k 上で run を列挙して、長さの条件を満たす square を数えればよい。
- run 列挙の実装によるが、計算量は全体で $O(n \log^2 n)$ または $O(n \log n)$ 時間。定数倍は大きめ。

解法3 (by hint908)

- `run` の列挙が不要な解法もある。
- A の後ろに $\text{neg}(A)$ を付け加えた、長さ $2n$ の数列を B とする。
- B に対し、接尾辞配列 (SA) と LCP 配列を構築しておく。
 - なお、蟻本の実装は負の数を含む入力を想定していないので注意。
- A が NC 列を含むならば、 B の前半と後半に共通連続部分列が出現する。
 - より具体的には、長さ $2m$ の列 $A[l..l+2m-1]$ が NC 列ならば、
 $B[l..l+m-1] = B[l+m+n..l+2m+n-1]$ となる。

1 2 -3 -4 -2 3 4 5 -1 -2 3 4 2 -3 -4 -5

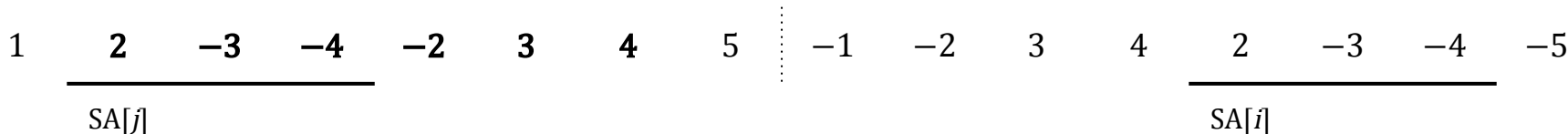
解法3 (by hint908)

- そのような、 B 中の共通連続部分列であって、長さや出現位置が適切なものを数えればよい。具体的には、 $i < j$ かつ

$$1 \leq |SA[i] - SA[j]| - n \leq \text{lcp}(B[SA[i]..], B[SA[j]..])$$

を満たす (i, j) の組を数えることにする。

- つまり、 $SA[i]$ と $SA[j]$ のうち、一方が前半、もう一方が後半に位置するものであって、それらの位置からの共通連続部分列長が十分に長いもの。



解法3 (by hint908)

- 不等式 $1 \leq |SA[i] - SA[j]| - n \leq \text{lcp}(B[SA[i]..], B[SA[j]..])$ を簡略化したい。
- LCP 配列の性質から、 $\text{lcp}(B[SA[i]..], B[SA[j]..]) = \min(\text{LCP}[i+1 .. j])$ である。
- LCP 配列に対する cartesian tree を考える。各節点 v について、その子孫からなる区間を $[l_v, r_v]$ とする。このとき、 $l_v - 1 \leq i < v \leq j \leq r_v$ を満たす範囲において、 $\min(\text{LCP}[i+1 .. j])$ は単に $\text{LCP}[v]$ になる。
- さらに、 i と j のうち一方が定数であれば、上記の不等式を満たす (i, j) の個数は、BIT または Wavelet 行列上のクエリとして書ける形になる。そこで、 $l_v - 1 \leq i < v$ と $v \leq j \leq r_v$ のうち、範囲が狭い方を選んでループを回すと、クエリの回数が全体で $O(n \log n)$ 回で抑えられる。
- 計算量は全体で $O(n \log^2 n)$ 時間。

ジャッジ解

● climpet(解法1)	C++	147 lines,	2.8kB
● climpet(解法2)	C++	144 lines,	2.8kB
● climpet(解法3)	C++	220 lines,	4.3kB
● hos	C++	272 lines,	8.4kB
● riantkb (chatgpt)	C++	158 lines,	3.0kB
● riantkb (chatgpt)	Python	107 lines,	1.8kB

統計情報

- AC teams
 - 3 + 3 teams
- First Acceptance
 - yawn in Gifu (37 min)