

ICPC模擬国内予選2021

E: 回転

原案: riantkb

問題文: climpet

データセット: tsutaj

解答: climpet, tsutaj, riantkb

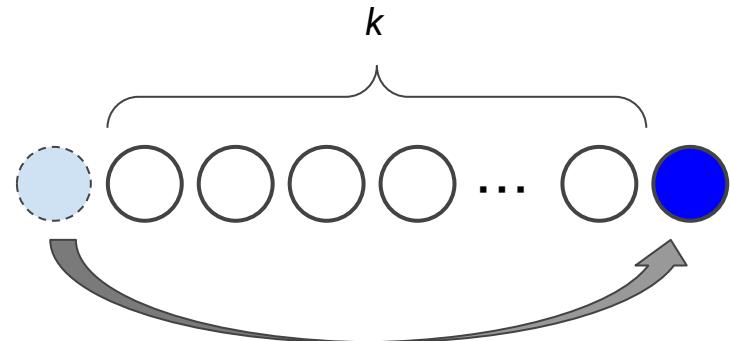
解説: climpet

問題概要

- 長さ N の順列 P が与えられる。
 - 「連續部分列を選んで回転させる」という操作を任意の回数行う。一回あたり、選んだ連續部分列の長さだけのコストがかかる。
 - 昇順ソートさせるのに必要なコストの合計を求めよ。
-
- 制約:
 - $1 \leq N \leq 10^5$

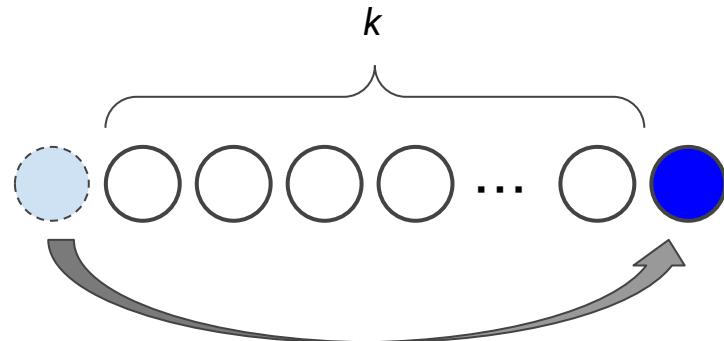
考察

- 選んだ連続部分列の長さを $k+1$ とする。
- 回転操作では、ある一要素の位置が、左または右に k だけ動く。それ以外の k 要素の位置は相対的に不变。
- したがって、回転とは、次のような操作であると解釈することができる。
 - コストを 1 支払い、要素を一つ選ぶ。さらにコストを k 支払い、選んだ要素の位置を k だけ左または右に動かす。



考察

- 順列 P の転倒数に着目すると、無駄な操作をしなければ、転倒数はちょうど k だけ減少する。すなわち、 k の合計は P の転倒数に等しい。
- あとは、各操作の最初にコスト 1 を支払うことになるので、操作回数を最小化すればよい。



考察

- ある要素を二回以上選ぶのは無駄なので、各要素は高々一回のみ選ぶと仮定して良い。
- 最適解において「一度も選ばない要素」を昇順に $U = (u_1, u_2, \dots, u_m)$ とする。
- これら m 要素の相対的な位置は不变なので、 U は P の（連続とは限らない）部分列でなければならない。
- 操作回数は $N - m$ 回となるので、操作回数を最小化するためには、 m を最大化する必要がある。
- すなわち、 U は P の最長単調増加部分列である。

まとめ

- 答えの下界として、次の値が得られる。
 - $(P \text{ の転倒数}) + N - (P \text{ の最長単調増加部分列長})$
- また、実際にコストの合計がこの下界と一致する操作列が構成可能である。すなわち、これが最適解である。
- 転倒数と最長単調増加部分列長はいずれも $O(N \log N)$ 時間で求められる。

ジャッジ解

- climpet (C++): 64行, 1076 bytes
- tsutaj(C++): 100行, 2756 bytes
- riantkb (Python): 63行, 1388 bytes

統計情報

- AC / trying teams
 - 45 / 73
- First acceptance
 - Heno World (21:01)