D: Self-Destruct Sequence

原案：hint908
問題文：riantkb
データセット：Ayuna
解答：Ayuna, hos, mtsd, riantkb
解説：riantkb

問題概要

- 長さ N の正整数列 A があたえられる
- A のすべての prefix に対して、以下の問題を答えよ
  - ある要素 A_i を選んで、その要素から始めて A_i 個の要素たちを削除する、という操作を繰り返す
    - A_i 個の要素がない場合は操作できない
  - これにより数列を空にできるかどうか判定、できるとき操作回数の最小値、最大値をそれぞれ求めよ

- 制約
  - N <= 5000

考察

- 空列にできるとき、最終的に各要素は以下のいずれかに割り当たる
  - 自分が選ばれて消される要素
  - 他の要素の選択によって一緒に消される要素

- 自分が選ばれて消される要素を決めたとき、
  それに対応するような操作列が存在する条件を考える

2 1 3 2 1 2

考察

- 前から見ながら、「いま何個消すことが予約されているか」を考える
- A_i を選ぶ場合
  - 削除予約が A_i - 1 だけ増える
- A_i を選ばない場合
  - 削除予約が 1 だけ減る

A	2	1	3	2	1	2	
削除予約	0	1	0	2	1	1	0

考察

- 削除予約が valid であるための条件は？

- すべての要素が非負
  - 負になるとその要素が残る
- 末尾が 0
  - これが正だと消しすぎる

- これが必要十分条件であることが示せる
  - 選んだ要素を後ろから順に消せるので、実際に可能な操作列が存在することが示せる

削除予約 0 1 0 2 1 1 0

解法

- よって、削除予約の値を管理しながらこれまでに選んだ要素数の最大と最小をそれぞれ管理すればよい
  - 前から DP することで、すべての prefix に対する答えを求めることができる
- 具体的には、dp[i + 1][j] = minmax(dp[i][j + 1], dp[i][j - (A[i] - 1)]) のような感じ
- 計算量は O(N^2)

ジャッジ解

- Ayuna C++ 33 lines, 842B
- hos C++ 83 lines, 2.3kB
- mtsd C++ 56 lines, 1.2kB
- riantkb C++ 30 lines, 658B
- riantkb Python 16 lines, 434B

統計情報

- AC teams
  - 20 + 1 teams
- First Acceptance
  - CCXX (18 min)