

JAG ICPC模擬国内予選2024

G: 数列の分割

原案 : mtsd

問題文 : climpet

データセット : mtsd

解答 : climpet, hos, mtsd, potato, riantkb

解説 : mtsd

問題概要

- 長さ N の数列 $A = (a_1, \dots, a_N)$ が与えられる
- 数列 A について、非空な連続部分列 B_1, \dots, B_m ($m \geq 1$, m : 任意) への分割を考える
 - つまり、 B_1, \dots, B_m の順に並べた列が A に一致する
- 分割スコア: $\sum (B_i \text{ の総和の2乗})$
- 全ての分割について、分割スコアを並べたときに、 K 番目に大きい値を答えよ
- 制約
 - $N \leq 1000$
 - $K \leq \min(2^{(N-1)}, 2000)$
 - $-10^6 \leq a_i \leq 10^6$

想定解: 動的計画法

- 以下のような動的計画法が考えられる
- $dp[n][x]$ = A の先頭 n 項(a_1, \dots, a_n) に対する x 番目に高い分割スコア
- $dp[n][1], \dots, dp[n][K]$ の候補になるのは以下の値
 - $dp[m][y] + (a_{m+1} + \dots + a_n)^2$ ($1 \leq m < n, 1 \leq y \leq K$)
 - ただし、 $dp[0][1] = 0$ とする
- 候補を全て計算して、上位 K 個をソートして取得する場合

計算量は $\Theta(N^2K \log(NK))$ になる (TLE)

想定解: 動的計画法

- $dp[n][x] = A$ の先頭 n 項(a_1, \dots, a_n) に対する x 番目に高い分割スコア
- 工夫: $dp[n][x]$ を更新する際には高々 n 個の候補のみを参照すればよい
 - $dp[m][*]$ については、まだ遷移で選ばれていないものの中で最大の候補のみを考えればよい

n	0	1	2	3	4	5
	$dp[0][1]=0$	$dp[1][1]$	$dp[2][1]$	$dp[3][1]$	$dp[4][1]$	$dp[5][1]$
		$dp[1][2]$	$dp[2][2]$	$dp[3][2]$	$dp[4][2]$	
		$dp[1][3]$	$dp[2][3]$	$dp[3][3]$	$dp[4][3]$	

図: $dp[5][1]$ の更新 (青色: 遷移の候補)

想定解: 動的計画法

- $dp[n][x] = A$ の先頭 n 項(a_1, \dots, a_n) に対する x 番目に高い分割スコア
- 工夫: $dp[n][x]$ を更新する際には高々 n 個の候補のみを参照すればよい
 - $dp[m][*]$ については、まだ遷移で選ばれていないものの中で最大の候補のみを考えればよい

n	0	1	2	3	4	5
	$dp[0][1]=0$	$dp[1][1]$	$dp[2][1]$	$dp[3][1]$ <small>best</small>	$dp[4][1]$	$dp[5][1]$
		$dp[1][2]$	$dp[2][2]$	$dp[3][2]$	$dp[4][2]$	$dp[5][2]$
		$dp[1][3]$	$dp[2][3]$	$dp[3][3]$	$dp[4][3]$	

図: $dp[5][2]$ の更新 (青色: 遷移の候補)

想定解: 動的計画法

- 優先度付きキューなどで管理して順に更新していけば、
dp[n][*] の更新は $O(n + K \log n)$ で実現可能である
- 全体の計算量は $O(N^2 + NK \log N)$

ジャッジ解

- climpet (C++) : 52 lines, 1024 bytes
- hos (C++) : 84 lines, 2325 bytes
- mtsd (C++) : 36 lines, 1023 bytes
- potato167(C++) : 37 lines, 1037 bytes
- riantkb (python) : 30 lines, 808 bytes

統計情報

- AC teams / Trying teams
 - 17 / 19
- First Acceptance
 - AMATSUKAZE (95:43)