

Balanced Segment Partition 1 - Solution Outline

Key Ideas

If we fix a maximum allowed segment sum, we can greedily split the array into the fewest segments. Feasibility is monotone, so binary search finds the minimal possible maximum.

Algorithm

1. Let $low = \max(a)$ and $high = \text{sum}(a)$.
2. While $low < high$:
 - Set $mid = \text{floor}((low + high) / 2)$.
 - Greedily scan the array, starting a new segment whenever adding the next value would exceed mid .
 - If the number of segments is at most k , set $high = mid$; otherwise set $low = mid + 1$.
3. Output low .

Correctness Sketch

Any valid split must have maximum segment sum at least $\max(a)$ and at most $\text{sum}(a)$. The greedy check produces the minimum number of segments for a given limit, so if it uses more than k , no split can satisfy that limit. This makes feasibility monotone in the limit, so binary search finds the smallest feasible value.

Complexity

- Time: $O(n \log(\text{sum}(a)))$
- Space: $O(1)$