

# Full Timeline Coverage 1 - Solution Outline

## Key Ideas

At each time, we should extend coverage as far as possible using clips that start at or before the current point. This is the same greedy idea as minimum jumps.

## Algorithm

1. Build an array `farthest[t]` storing the farthest end among clips that start at `t`.
2. Scan `time` from 0 to `T - 1`:
  - Update `bestReach = max(bestReach, farthest[time])`.
  - If `bestReach <= time`, return -1 (cannot move forward).
  - If `time == currentEnd`, increment the answer and set `currentEnd = bestReach`.
3. Output the answer.

## Correctness Sketch

At any point, to minimize the number of clips, we must extend the current coverage as far as possible using all clips that start before the current boundary. If we do not advance past the current time, coverage is impossible. This greedy strategy yields the minimum number of clips.

## Complexity

- Time:  $O(n + T)$
- Space:  $O(T)$