G266 Full Speed Ahead

Darren Lu {firewater}

2026-02-14

Background

Problem Idea by `kctung`

Preparation by `__jk__`

Presentation by `firewater`

Problem Statement

- Given four integers $1 <= D <= 10^{18}$, $0 <= S_0 <= 10^9$, $1 <= A <= 10^9$, and $\max(S_0, A) <= M <= 10^9$ in input.
- Find the smallest positive integer T for which there exists an array $[S_1, S_2, ..., S_T]$ satisfying all of the following conditions:
  - $0 <= S_i <= M$ for every $1 <= i <= T$
  - $|S_i - S_{i-1}| <= A$ for every $1 <= i <= T$
  - $S_T <= A$
  - $\sum_i S_i = D$
- Output -1 if such T does not exist.

Statistics

sosad

8: 6	14: 0	20: 0	19: 0	21: 0	18: 0

Subtask 1 (8%): M = A

- No need to care about acceleration limit, and car is always able to stop immediately after the drive.
- Drive as fast as possible (at speed M) unless the sum overshoots past D.
- So the answer is simply ceil(D / M).

Subtask 2 (14%): S_0 = M, A = 1, D and M are small

- Car starts at maximum possible speed, can accelerate or decelerate at most 1 per hour.
- Consider a greedy strategy, where the speeds at each hour are some (possibly zero) number of M-s, followed by an arithmetic sequence with common difference -1 from M - 1 down to 1: [M, ..., M, M - 1, M - 2, ..., 1].
- If such a sequence with sum D exists (so $x * M + M * (M - 1) / 2 = D$ has a non-negative integer solution), then it gives an optimal solution.
- But if x is negative, then the sum must be too large, so output -1.
- And if x is non-integer, then use floor(x) copies of M, and insert the remainder into the decreasing part. It is also to see that this is optimal.

Subtask 3 (20%): A = 1, D and M are small

- It is not difficult to observe that the following greedy strategy works: iterate from i = 0 increasingly, and suppose the speed is S_i at the i-th second, then S_{i+1} is the largest integer such that:
  - $\max(0, S_i - A) <= S_{i+1} <= \min(M, S_i + A)$, and
  - Putting S_{i+1} does not force the total sum to overshoot D, so $S_1 + S_2 + ... + S_i + S_{i+1} + S_{i+1} * (S_{i+1} - 1) / 2 <= D$. (by Subtask 2)
- Doing this greedy naively, the time complexity is O(answer * A).
- Since the answer is bounded above by D, this fits within the time limit.

Subtask 4 (19%): $S_0 = 0$

- We could modify the greedy from Subtask 3, but it would still be too slow.
- Let's formulate an alternative problem instead:

  Problem: for a fixed constant time limit T, and given S_0, M, and A in input, what are the distances possible to be reached in (at most) T seconds?
- Clearly, such reachable distances form an interval $[0, D_T]$, and so we wish to find the value of D_T for each T.
- **Observation:** $D_T <= D_{T+1}$ for all T.
- So, once we know how to find the values of D_T, we can simply perform a binary search to obtain the answer.

Subtask 4 (19%): $S_0 = 0$

- Fix T. Clearly, S_i <= min(M, min(i, T - i + 1) * A) for every 1 <= i <= T.
- **Observation:** the sequence S_i = min(M, min(i, T - i + 1) * A) is valid.
- This means our desired D_T is simply ∑ min(M, min(i, T - i + 1) * A).
- This can be calculated in O(1) time:
  - If T <= 2 * floor(M / A), then it is an A.S. with d = A with ceil(T / 2) terms, then another A.S. with d = -A with floor(T / 2) terms.
  - If T > 2 * floor(M / A), then it is an A.S. with d = A with floor(M / A) terms, then T - 2 * floor(M / A) terms of M, then another A.S. with d = -A with floor(M / A) terms.
- This solves subtasks 1, 3, and 4 together. (Score: 8 + 20 + 19 = 47)

Subtasks 5 (21%) and 6 (18%): Full Solution

- Similar to Subtask 4, we ask the same problem again.
- However, the reachable distances are no longer intervals $[0, D_T]$ as in Subtask 4. Nevertheless, we can still note the following:
- **Observation:** the reachable distances form a (possibly empty) interval $[D_T^{min}, D_T^{max}]$ for each fixed time limit T.
- Clearly, if $ceil(S_0 / A) - 1 > T$, then there is no reachable distance
- Otherwise, $D_T^{min} = \sum_i S_0 - i * A$, where the sum ends right before the terms become negative. Therefore, we can deduce that:
- **Observation:** D_T^{min} and D_T^{max} are each monotonically increasing (for the non-empty intervals).

Subtasks 5 (21%) and 6 (18%): Full Solution

- Similar as Subtask 4, once we fix T $>= ceil(S_0 / A) - 1$ (so that we guarantee a construction exists), we must have that for every 1 <= i <= T, $S_i <= min(M, S_0 + i * A, (T - i + 1) * A)$ holds.
- Also, we can make the same observation, that when we take the equality case above for every 1 <= i <= T, it forms a valid sequence.
- The desired sum $D_T^{max} = \sum min(M, S_0 + i * A, (T - i + 1) * A)$ can also be calculated in O(1) time. Separate cases for which of the three expressions are attaining the minimum. Details are left as an exercise for the reader.
- So we can do binary search just like in Subtask 4. Remember to handle the -1 case. This solves all subtasks.