Week 6 PSO Questions

TAs

Mukul Agarwal Aaryan Wadhwani

September 29, 2026

Mountain Getaway. A robber is trying to escape a mountainous grid of size $n \times m$, represented as a two-dimensional array $H[i, j]$, where $H[i, j]$ is the elevation in feet at (i, j). The robber starts at cell $(1, 1)$ and wants to reach the getaway point (n, m). On each move, he can move one cell right or one cell down.

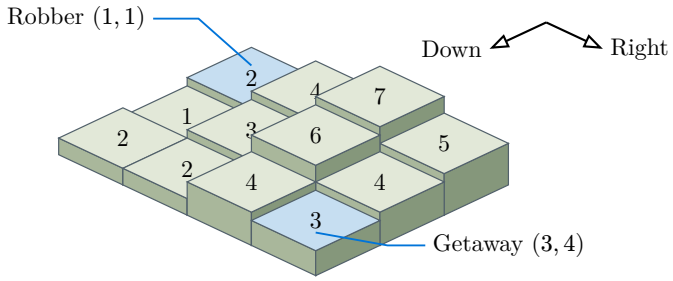


Figure 1: Example of a mountainous grid.

Because he is running on foot, the robber can climb by at most K_R feet in a single move.[1] Downhill movement is not restricted.

1. You are a police officer who has a human-sized mousetrap on one of the cells other than $(1, 1)$ or (n, m) before the robber starts his journey. The robber knows which cell is blocked and can try to choose any legal route around it. Design and analyze an algorithm (the faster the better) to determine if you can prevent the robber from reaching his destination.
2. In a more realistic scenario, we instead have the police officer giving chase from $(n, 1)$ while the robber starts at $(1, 1)$. Each minute, both *must* move one cell: the robber right or down, the officer right or up. The officer's climbing gear allows climbs of at most K_P feet per move.[2]

[1] That is, a robber cannot move from (i, j) to (i', j') if $H[i', j'] > H[i, j] + K_R$.

[2] We hope (but don't assume) that $K_P > K_R$, since the police probably have a bigger equipment budget.

If the officer and the robber know where the other is at all times, can the officer guarantee catching the robber before the robber reaches (n, m)?

Going in Circles. The robber has escaped the grid and is now driving clockwise around a circular mountain road with N checkpoints. You know the elevation $A[i]$ of each checkpoint i, numbered clockwise from 1 to N. Checkpoint 1 comes immediately after checkpoint N.

Fortunately, the police planted an altitude tracker in the getaway car. Unfortunately, they bought the cheapest one available: every reading is off by the same unknown constant c. The tracker reports a sequence $B[1..k]$ from k consecutive checkpoints, where $1 \le k \le N$, and the robber is currently at the checkpoint that produced the last reading. The tracker does not skip any checkpoints.

For example, suppose the checkpoint elevations in clockwise order are $(10, 14, 11, 20, 24, 21)$ and the readings are $(100, 104, 101)$. The robber could currently be at checkpoint 3 or checkpoint 6: the unknown offset could be 90 or 80, respectively.

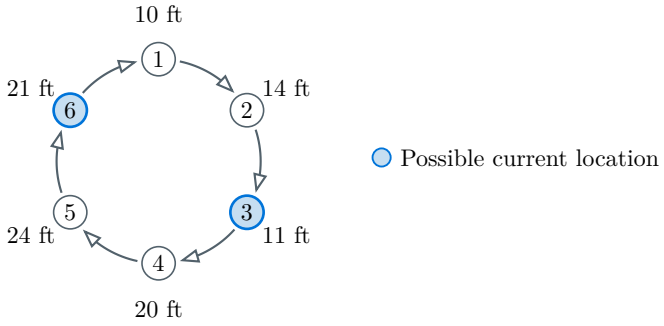


Figure 2: Example of checkpoints $(10, 14, 11, 20, 24, 21)$ with readings $(100, 104, 101)$.

Design and analyze an algorithm that returns every checkpoint where the robber could currently be (the faster, the better). Remember that the robber may have crossed from checkpoint N to checkpoint 1.