P2

Prompt:

Given the described wall, the algorithm returns the smallest number of steps to collect the coins.

Algorithm:

The problem doesn't specify a format so we assume the following.

Let the wall be represented as a list of m points P = [p_0, ..., p_m] that are visited such that p_0 is the start point (0, 0), p_1 is the point of the first coin, p_2 the point of the second, ..., p_{m-2} is the point of the second to last coin, p_{m-1} is the point of the last coin, and p_m is the point (k, n). In this format, p_1 and p_2 are lowest coins on the wall, p_3 and p_4 are on the next row up, ..., and so on up to p_{m-2} and p_{m-1} which are the highest coins on the wall.

Let d(i, j) be the distance function that takes points i and j and returns the minimum number of legal steps: left, right, and up ladders, between i and j.

```
Input: The list of points P to visit on the wall
Output: The smallest number of steps to collect the coins
MinStep(P):
    Let V be an empty list
    V[1] = d(p_0, p_2) + d(p_2, p_1)
    V[2] = d(p_0, p_1) + d(p_1, p_2)
    For indices 2 < i < m of P do
        If i is odd then
            V[i] = d(p_{i+1},p_i) + min{V[i-1] + d(p_{i-1},p_{i+1}), V[i-2] + d(p_{i-2},p_{i+1})}
        Else if i is even then
            V[i] = d(p_{i-1},p_i) + min{V[i-2] + d(p_{i-2},p_{i-1}), V[i-3] + d(p_{i-3},p_{i-1})}
        EndIf
    V[m] = min{V[m-1] + d(p_{m-1},p_i), V[m-2] + d(p_{m-2}, p_m)}
    Return V[m]
```

Claim:

The algorithm terminates in time polynomial to the width and height of the wall.

Proof:

That the algorithm terminates is obvious given that each element in the array is accessed once and only once. The number of steps to reach each point is cached as an array element as it's found, so during loop iteration, previously found step costs are accessed in constant time. Therefore, the dominant runtime factor is the loop. Thus, the algorithm terminates in polynomial time. □

Claim:

The algorithm outputs the smallest number of steps to collect all the coins.

Proof:

Let OPT(i) denote the smallest number of steps to visit points $1 \leq i \leq m$. With 0 as the start point, i=1 is the first coin, i=2 is the second coin, ..., i=m-1 is the last coin, and i=m is the end point at (k,n). There are a pair of coins on each row of the wall. Therefore, each pair consists of a coin at an even numbered point and an odd numbered point. We can say then that for any coin i the coin in the same row that i is paired with is either at i-1 if i is even or i+1 if i is odd. Also note, that for any coin $1 \leq i < m$, OPT(i) always assumes the other coin on the same row was visited first.

Case 1: If i=1 then OPT(i) assumes the first even numbered coin i+1 was visited first. Thus, the minimum steps to i is the shortest path from the start point 0 to the second coin i+1 to the first coin i.

$$\text{OPT}(1) = \text{d}(0, \text{i+1}) + \text{d}(\text{i+1}, \text{i})$$

Case 2: If i=2 then OPT(i) assumes the first odd numbered coin i-1 was visited first. Thus, the minimum steps to i is the shortest path from the start point 0 to the first coin i-1 to the second coin i.

$$\text{OPT}(2) = \text{d}(0, \text{i-1}) + \text{d}(\text{i-1}, \text{i})$$

Case 3: If $2<i<m$ and i is odd then OPT(i) assumes the even numbered coin i+1 in the same row was visited first. The minimum steps between i and i+1 are d(i, i+1). Now we need to decide on the minimum step path p that led to i+1. We know that p will be from a coin in the previous row, either from the even numbered coin i-1 or the odd numbered coin i-2. Let the minimum step path to i+1 from i-1 be p_{i-1}. Then p_{i-1} will be the minimum steps d(i-1, i+1) between i-1 and i+1, plus the minimum step path OPT(i-1) from the start to i-1. Let the minimum step path to i+1 from i-2 be p_{i-2}. Then p_{i-2} will be the minimum steps d(i-2, i+1) between i-2 and i+1, plus the minimum step path OPT(i-2) from the start to i-2. We let p be the minimum of p_{i-1} and p_{i-2}. Thus, when $2<i<m$ and i is odd, the shortest path to i from the start through the even numbered coin i+1 in the same row will be:

$$\text{OPT}(\text{i}) = \text{d}(\text{i+1}, \text{i}) + \min\{\text{OPT}(\text{i-1}) + \text{d}(\text{i-1}, \text{i+1}), \text{OPT}(\text{i-2}) + \text{d}(\text{i-2}, \text{i+1})\}$$

Case 4: If $2<i<m$ and i is even then OPT(i) assumes the even numbered coin i-1 in the same row was visited first. The minimum steps between i and i-1 are d(i-1, i). Now we need to decide on

the minimum step path p that led to i-1. We know that p will be from a coin in the previous row, either from the even numbered coin i-2 or the odd numbered coin i-3. Note that the minimum even i>2 is i=4 so i-3=1 which is a base case. Which is to say, that i-3 doesn't take us out of bounds below our lowest base case i=1. Let the minimum step path to i-1 from i-2 be p_{i-2}. Then p_{i-2} will be the minimum steps d(i-2, i-1) between i-2 and i-1, plus the minimum step path OPT(i-2) from the start to i-2. Let the minimum step path to i-1 from i-3 be p_{i-3}. Then p_{i-3} will be the minimum steps d(i-3, i-1) between i-3 and i-1, plus the minimum step path OPT(i-3) from the start to i-3. We let p be the minimum of p_{i-2} and p_{i-3}. Thus, when 2<i<m and i is even, the shortest path to i from the start through the odd numbered coin i-1 in the same row will be:

$$\mathrm{OPT(i)} = \mathrm{d(i\text{-}1, i)} + \min\{\mathrm{OPT(i\text{-}2)} + \mathrm{d(i\text{-}2, i\text{-}1)}, \mathrm{OPT(i\text{-}3)} + \mathrm{d(i\text{-}3, i\text{-}1)}\}$$

Case 5: If i=m, then by definition i is the last point on the wall, (k,n). Consider that the minimum step path p to i from the start must be through a coin on the last row, either coin i-1 or coin i-2. Let the minimum step path to i from i-1 be p_{i-1}. Then p_{i-1} will be the minimum steps d(i-1, i) between i-1 and i, plus the minimum step path OPT(i-1) from the start to i-1. Let the minimum step path to i from i-2 be p_{i-2}. Then p_{i-2} will be the minimum steps d(i-2, i) between i-2 and i, plus the minimum step path OPT(i-2) from the start to i-2. We let p be the minimum of p_{i-1} and p_{i-2}. Thus, the shortest path to i from the start is:

$$\mathrm{OPT(i)} = \min\{\mathrm{OPT(i\text{-}1)} + \mathrm{d(i\text{-}1, i)}, \mathrm{OPT(i\text{-}2)} + \mathrm{d(i\text{-}2, i)}\}$$

Thus, our recurrence relation for OPT(i):

$$\mathrm{OPT(i)} = \begin{cases} \mathrm{d(0, i\text{+}1)} + \mathrm{d(i\text{+}1, i)} & \text{if i=1} \\ \mathrm{d(0, i\text{-}1)} + \mathrm{d(i\text{-}1, i)} & \text{if i=2} \\ \mathrm{d(i\text{+}1, i)} + \min\{\mathrm{OPT(i\text{-}1)} + \mathrm{d(i\text{-}1, i\text{+}1)}, \mathrm{OPT(i\text{-}2)} + \mathrm{d(i\text{-}2, i\text{+}1)}\} & \text{if } 2 < i < n \text{ and i is odd} \\ \mathrm{d(i\text{-}1, i)} + \min\{\mathrm{OPT(i\text{-}2)} + \mathrm{d(i\text{-}2, i\text{-}1)}, \mathrm{OPT(i\text{-}3)} + \mathrm{d(i\text{-}3, i\text{-}1)}\} & \text{if } 2 < i < n \text{ and i is even} \\ \min\{\mathrm{OPT(i\text{-}1)} + \mathrm{d(i\text{-}1, i)}, \mathrm{OPT(i\text{-}2)} + \mathrm{d(i\text{-}2, i)}\} & \text{if i=n} \end{cases}$$

□