

donkey

October 15, 2024

1 INSTRUCTIONS

1. please copy the code below and append it on the code provided in the CSA Academy
2. Link to the challenge here: <https://csacademy.com/contest/interview-archive/task/donkey-paradox/statement/>
3. Link to the slides here: <https://the-donkey-paradox-a-gri-d2nknew.gamma.site/>

1.1 STARTER CODE:

This piece of code is meant to help you get the inputs required for you to solve the challenge

```
[ ]: N = get_number()
M = get_number()
# M = get_number()

# Read and parse the first haystack coordinates into a tuple
haystack1 = (get_number() - 1, get_number() - 1) # Convert to 0-based index

# Read and parse the second haystack coordinates into a tuple
haystack2 = (get_number() - 1, get_number() - 1) # Convert to 0-based index
```

1.2 COMPLETE SOLUTION

This is my implementation of the challenge, I hope the comments left therein are helpful

```
[ ]: def count_donkey_starvation_cells(N, M, haystack1, haystack2):
    # N X M = our matrix - grid
    # Extract haystack position
    hx1, hy1 = haystack1
    hx2, hy2 = haystack2

    # Counter for starvation cells - what is our end goal? return the total
    ↪ number of cells where our donkey will starve
    starvation_count = 0

    # Iterate over all cells in the matrix
    for i in range(N):
        for j in range(M):
            # Calculate Manhattan distance to both haystacks
```

```

        # Why Manhattan? - remember the donkey can only go two ways?
        ↪ Horizontal & vertical?
        dist1 = abs(i - hx1) + abs(j - hy1)
        dist2 = abs(i - hx2) + abs(j - hy2)

        # If distances are equal, this is a starvation cell : This is
        ↪ really what we care about
        if dist1 == dist2:
            starvation_count += 1

    # Output the result
    return starvation_count

N = get_number()
M = get_number()
# M = get_number()

# Read and parse the first haystack coordinates into a tuple
haystack1 = (get_number() - 1, get_number() - 1) # Convert to 0-based index

# Read and parse the second haystack coordinates into a tuple
haystack2 = (get_number() - 1, get_number() - 1) # Convert to 0-based index

# Get the result = I want you to implement the count_donkey_starvation_cells
result = count_donkey_starvation_cells(N, M, haystack1, haystack2)

# Print the number of starvation cells
print(result)

```