1066. Campus Bikes II Premium

Solved

Medium · Topics · Companies · Hint

On a campus represented as a 2D grid, there are `n` workers and `m` bikes, with `n <= m`. Each worker and bike is a 2D coordinate on this grid.

We assign one unique bike to each worker so that the sum of the **Manhattan distances** between each worker and their assigned bike is minimized.

Return `the minimum possible sum of Manhattan distances between each worker and their assigned bike`.

The **Manhattan distance** between two points `p1` and `p2` is `Manhattan(p1, p2) = |p1.x - p2.x| + |p1.y - p2.y|`.

Bit Mask + Memorization (DP)

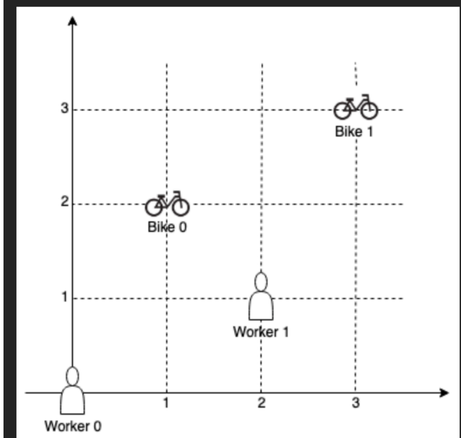
↓

to track which bike is used

E.g. have 3 bikes.

mask =	0b000	used bike ∅
	0b100	b[2]
	0b011	b[0], b[1]

Example 1:




```
Input: workers = [[0,0],[2,1]], bikes = [[1,2],[3,3]]
Output: 6
Explanation:
We assign bike 0 to worker 0, bike 1 to worker 1. The Manhattan distance of both
assignments is 3, so the output is 6.
```

python

```python
from functools import lru_cache

def assignBikes(workers, bikes):
    n, m = len(workers), len(bikes)

    @lru_cache(None)
    def dp(i, mask):
        if i == n:
            return 0
        res = float('inf')
        for j in range(m):
            if not (mask & (1 << j)):
                dist = abs(workers[i][0] - bikes[j][0]) + abs(workers[i][1] - bikes[j][1])
                res = min(res, dist + dp(i + 1, mask | (1 << j)))
        return res

    return dp(0, 0)
```

@lru_cache(None) → LRU : when same subproblem comes again, avoid recomputation

mask | (1 << j) : set bit j to 1

1<<j : shift 1 by j bits

mask & (1<<j) : check if j-th bit in mask is 1.

if not(..) : run only if j-th mask is 0, which means the bike[j] is still available