Minimum Path Sum

Submission Detail

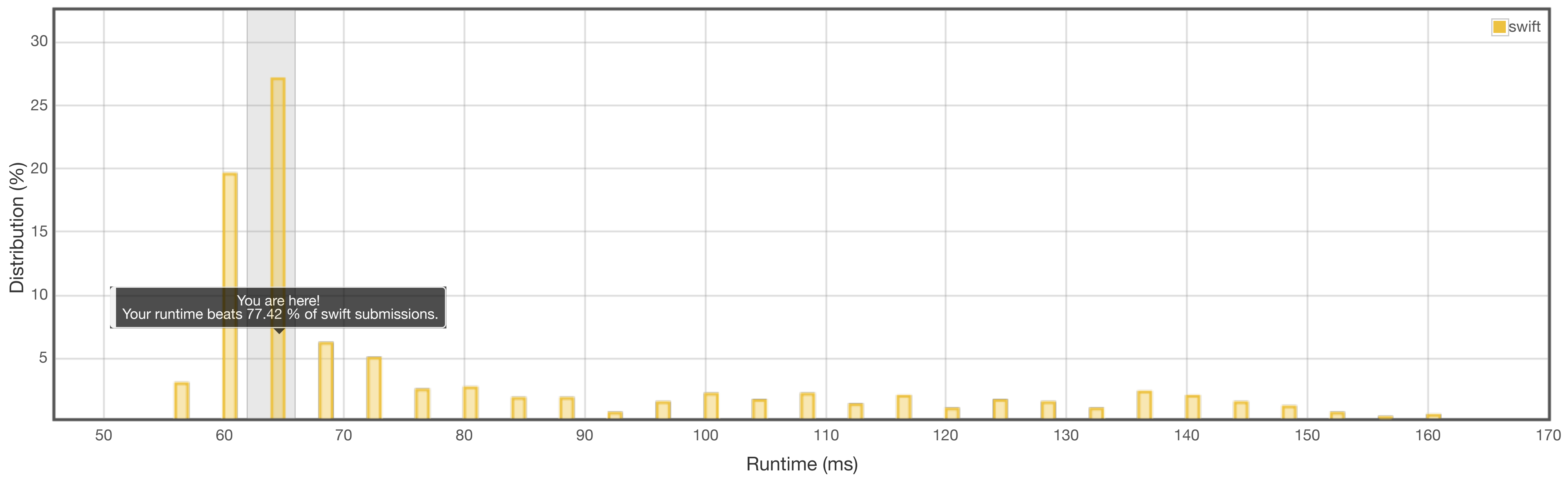
61 / 61 test cases passed.

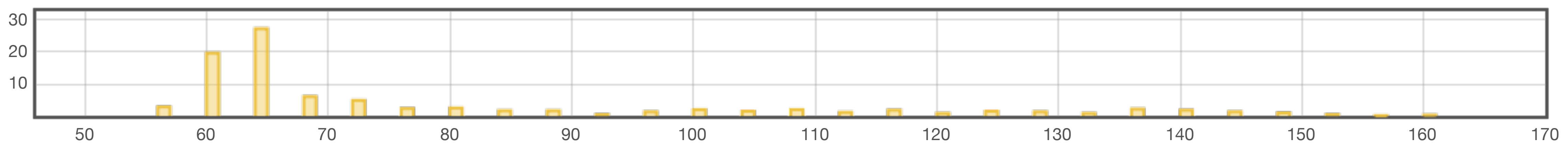
Runtime: **64 ms**
Memory Usage: **21.3 MB**

Status: Accepted

Submitted: **0 minutes ago**

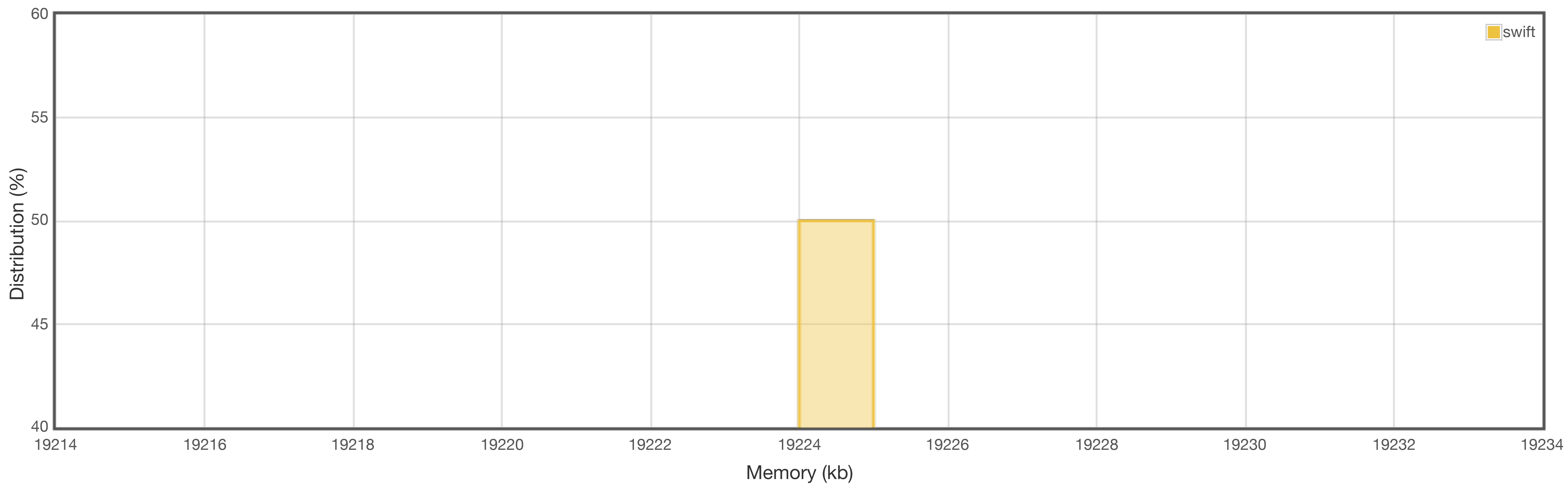
Accepted Solutions Runtime Distribution

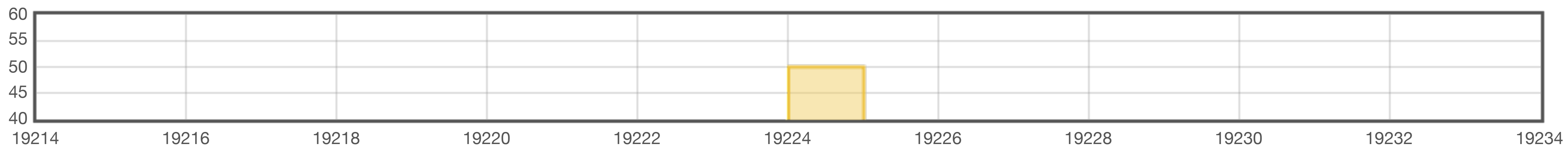




Zoom area by dragging across this chart

Accepted Solutions Memory Distribution





Zoom area by dragging across this chart

Invite friends to challenge **Minimum Path Sum**

20

Submitted Code: 0 minutes ago

Language: swift

Edit Code

```swift
class Solution {
    var m = 0
    var n = 0
    var dp: [[Int]] = [[]]

    func getDp(_ i: Int, _ j: Int) -> Int? {
        guard i >= 0 else { return nil }
        guard j >= 0 else { return nil }
        return dp[i][j]
    }

    func minPathSum(_ grid: [[Int]]) -> Int {
        n = grid.count
        guard n > 0 else { return 0 }
        m = grid[0].count

        dp = Array.init(repeating: Array.init(repeating: 0, count: m), count: n)
        var i = 0
        while i < n {
            var j = 0
            while j < m {
                let left = getDp(i, j-1)
                let up = getDp(i-1, j)

                if let left = left, let up = up {
                    dp[i][j] = Swift.min(left, up) + grid[i][j]
                } else if let left = left {
                    dp[i][j] = left + grid[i][j]
                } else if let up = up {
                    dp[i][j] = up + grid[i][j]
                } else {
                    dp[i][j] = grid[i][j]
                }

                j += 1
            }
            i += 1
        }

        return dp[n-1][m-1]
    }
}
```

Back to problem