Do:

$\forall\{A, B \subseteq Rods\}$ where $A \cap B = \emptyset$, update a dictionary which its keys are represented by $h_{\mathrm{diff}} = \left|\sum_{a \in A} a - \sum_{b \in B} b\right|$.

1. Add rod to the taller one

In this case, the shorter counterpart doesn't change at all so we should update using that counterpart.

```
_dp[h_diff + h] = max(_dp[h_diff + h], _h)
```

2. Add rod to the shorter one

There are two cases in this scenario:

1. If $h_{\mathrm{diff}} \geq h_{\mathrm{rod}}$, the height difference will shrink to $h_{\mathrm{diff}} := h_{\mathrm{diff}} - h_{\mathrm{rod}}$ while shorter one remains the shorter position. Therefore we should update with the present height of the shorter rod.
2. If $h_{\mathrm{diff}} \leq h_{\mathrm{rod}}$, two rods exchange there positions; taller to shorter, shorter to taller. We therefore update with the height of the previous taller rod: previous shorter rod + height difference

```
if h_diff >= h:
    _dp[h_diff - h] = max(_dp[h_diff - h], _h + h)
else:
    _dp[h - h_diff] = max(_dp[h - h_diff], _h + h_diff)
```

Code

```
def tallestBillboard(self, rods: List[int]) -> int:
    dp = defaultdict(int)
    dp[0] = 0

    for h in rods:
        _dp = dp.copy()
        for h_diff, _h in dp.items():
            _dp[h_diff + h] = max(_dp[h_diff + h], _h)
            if h_diff >= h:
                _dp[h_diff - h] = max(_dp[h_diff - h], _h + h)
            else:
                _dp[h - h_diff] = max(_dp[h - h_diff], _h + h_diff)
        dp = _dp
    return dp[0]
```