lee215 ★ 47714 Last Edit: October 19, 2019 10:04 PM 1.9K VIEWS

120

Please reply and upvote rather than later.
Don't have prime membership.
Can't even read and modify my own post later, when it's locked.

Explanation

`dp[c][k]` is the prob of tossing `c` first coins and get `k` faced up.
`dp[c][k] = dp[c - 1][k] * (1 - p) + dp[c - 1][k - 1] * p)`
where `p` is the prob for `c` -th coin.

Complexity

Time `O(N^2)`
Space `O(N)`

Java:

```
public double probabilityOfHeads(double[] prob, int target) {
    double[] dp = new double[target + 1];
    dp[0] = 1.0;
    for (int i = 0; i < prob.length; ++i)
        for (int k = Math.min(i + 1, target); k >= 0; --k)
            dp[k] = (k > 0 ? dp[k - 1] : 0) * prob[i] + dp[k] * (1 - prob[i]);
    return dp[target];
}
```

C++:

```
double probabilityOfHeads(vector<double>& prob, int target) {
    vector<double> dp(target + 1);
    dp[0] = 1.0;
    for (int i = 0; i < prob.size(); ++i)
        for (int k = min(i + 1, target); k >= 0; --k)
            dp[k] = (k ? dp[k - 1] : 0) * prob[i] + dp[k] * (1 - prob[i]);
    return dp[target];
}
```

Python:

```
def probabilityOfHeads(self, prob, target):
    dp = [1] + [0] * target
    for p in prob:
        for k in xrange(target, -1, -1):
            dp[k] = (dp[k - 1] if k else 0) * p + dp[k] * (1 - p)
    return dp[target]
```

Comments: 9 Best | Most Votes | Newest to Oldest | Oldest to Newest

Type comment here... (Markdown is supported)

Post

lanfker ★ 61 October 19, 2019 10:18 PM

i thought of DP as well, but i used a 2D array. Love your sln. For anyone who is interested in 2D array DP.. LOL.

```
class Solution {
public:
        //initialize choosing no heads at all...
        F[0][0] = 1-prob[0];
        F[0][1] = prob[0];
```

Read More

9 Show 4 replies Reply

Ivan666 ★ 405 October 19, 2019 9:43 PM

Brilliant solution, I got TLE for dfs solutions. I just cannot figure out a dp solution. Hate myself

5 Show 3 replies Reply

ethan-marsh ★ 40 Last Edit: October 20, 2019 5:32 AM

For anyone who wants more explanation, check out **Binomial Distribution - Wikipedia.**

4 Show 2 replies Reply

shuuchen ★ 142 Last Edit: October 27, 2019 7:39 PM

I wish I can write in 1D, but I can only understand 2D.
my dp solution in 2D array in python

```
class Solution:
    def probabilityOfHeads(self, P: List[float], target: int) -> float:
```

Read More

1 Show 1 reply Reply

Hong_tao ★ 123 October 20, 2019 3:11 PM

orz

1 Reply

shuheik ★ 32 March 7, 2020 8:25 AM

Similar idea, but slightly better runtime with `min(i + 2, target + 1)` trick.

```
from collections import defaultdict

class Solution:
    def probabilityOfHeads(self, prob: List[float], target: int) -> float:
        dp = defaultdict(int)

        dp[(0, 0)] = 1 - prob[0]
        dp[(0, 1)] = prob[0]
```

Read More

0 Reply

ddoudle ★ 798 February 23, 2020 12:17 AM

Same idea:

```
class Solution:
    def probabilityOfHeads(self, prob: List[float], target: int) -> float:
        memo = {}

        def subprob(end, t):
            if (end,t) in memo:
                return memo[(end,t)]
            if end <= t-2:
                return 0
```

Read More

0 Reply Share Report

forestsong ★ 182 October 23, 2019 8:50 AM

Great solution! Optimized a little bit:

```
class Solution:
    def probabilityOfHeads(self, prob: List[float], target: int) -> float:
        if target > len(prob): return 0
        dp = [1] + [0] * target
        for i, p in enumerate(prob):
            for k in range(min(i+1, target), -1, -1):
                dp[k] = (dp[k - 1] if k else 0) * p + dp[k] * (1 - p)
        return dp[target]
```

0 Reply

Merciless ★ 489 October 19, 2019 9:47 PM

Similar idea, less intuitive

```
class Solution:
    def probabilityOfHeads(self, prob: List[float], target: int) -> float:
        dp = {0 : 1}
        for i in prob:
            ndp = collections.Counter()
            for k,v in dp.items():
                ndp[k + 1] += i * v
                ndp[k] += (1 - i) * v
            dp = {i:j for i,j in ndp.items() if i <= target}
```

Read More

0 Show 1 reply Reply