

[Description](#) | [Solution](#) | [Submissions](#)[Discuss \(33\)](#)[Back](#) | [Java/C++/Python] DP and O(N)| [Upvote](#) [Downvote](#) [Comment](#)

lee215

★ 47713

Last Edit: November 16, 2019 11:44 PM 1.3K VIEWS

23

**Solution 1: DP**

`dp[n]` is the number of shaking ways of  $n$  pairs people  
 In the view of first people in these  $n$  pairs,  
 he/she can choose anyone, split  $i$  pairs on his left and  $n - 1 - i$  pairs on his right.

So here comes the equation of dynamic programme:

$$dp[n + 1] = dp[0] * dp[n] + dp[1] * dp[n - 1] + \dots + dp[n] * dp[0]$$
**Complexity**Time  $O(N^2)$ Space  $O(N)$ **Java:**

```
public int numberOfWays(int n) {
    long mod = (long)1e9 + 7;
    long[] dp = new long[n / 2 + 1];
    dp[0] = 1;
    for (int k = 1; k <= n / 2; ++k) {
        for (int i = 0; i < k; ++i) {
            dp[k] = (dp[k] + dp[i] * dp[k - 1 - i]) % mod;
        }
    }
    return (int)dp[n / 2];
}
```

**C++:**

```
int numberOfWays(int n) {
    long mod = 1e9 + 7;
    vector<long> dp(n / 2 + 1);
    dp[0] = 1L;
    for (int k = 1; k <= n / 2; ++k) {
        for (int i = 0; i < k; ++i) {
            dp[k] = (dp[k] + dp[i] * dp[k - 1 - i]) % mod;
        }
    }
    return dp[n / 2];
}
```

```
def numberOfWays(self, n):
    return sum(self.numberOfWays(i) * self.numberOfWays(n - 2 - i) for i in range(0, n, 2)) if n else 1
```

**Solution 2: Catalan Numbers**

Inspired from @awice

**Java**

```
public int numberOfWays(int n) {
    long[] inv = new long[n / 2 + 2];
    inv[1] = 1;
    long mod = (long)1e9 + 7, res = 1;
```

```
for (int i = 2; i < n / 2 + 2; ++i) {
    inv[i] = mod - mod / i * inv[(int)mod % i] % mod;
}
```

```
for (int i = 1; i <= n / 2; ++i) {
    res = res * (i + n / 2) % mod;
```

```
res = res * inv[i] % mod;
}
```

```
return (int)(res * inv[n / 2 + 1] % mod);
}
```

**C++**

```
int numberOfWays(int n) {
    vector<long> inv(n / 2 + 3);
    inv[1] = 1;
    long mod = 1e9 + 7, res = 1;
```

```
for (int i = 2; i < n / 2 + 2; ++i) {
    inv[i] = mod - mod / i * inv[mod % i] % mod;
}
```

```
for (int i = 1; i <= n / 2; ++i) {
    res = res * (i + n / 2) % mod;
```

```
res = res * inv[i] % mod;
}
```

```
return res * inv[n / 2 + 1] % mod;
}
```

**Python**

```
def numberOfWays(self, n):
    res = 1
    for i in xrange(1, n / 2 + 1):
        res *= n - i + 1
        res /= i
    return res / (n / 2 + 1) % (10**9 + 7)
```

Comments: 1

Best Most Votes Newest to Oldest Oldest to Newest



kusirroll ★ 133 November 20, 2019 3:03 PM

I thought I could do this with a closed form expression during the contest....

0

Reply

Post