1646. Get Maximum in Generated Array

You are given an integer `n`. An array `nums` of length `n + 1` is generated in the following way:

- `nums[0] = 0`
- `nums[1] = 1`
- `nums[2 * i] = nums[i]` when `2 <= 2 * i <= n`
- `nums[2 * i + 1] = nums[i] + nums[i + 1]` when `2 <= 2 * i + 1 <= n`

Return*the maximum integer in the array *`nums`.

Example 1:

```
Input: n = 7
Output: 3
Explanation: According to the given rules:
  nums[0] = 0
  nums[1] = 1
  nums[(1 * 2) = 2] = nums[1] = 1
  nums[(1 * 2) + 1 = 3] = nums[1] + nums[2] = 1 + 1 = 2
  nums[(2 * 2) = 4] = nums[2] = 1
  nums[(2 * 2) + 1 = 5] = nums[2] + nums[3] = 1 + 2 = 3
  nums[(3 * 2) = 6] = nums[3] = 2
  nums[(3 * 2) + 1 = 7] = nums[3] + nums[4] = 2 + 1 = 3
Hence, nums = [0,1,1,2,1,3,2,3], and the maximum is 3.
```

Example 2:

```
Input: n = 2
Output: 1
Explanation: According to the given rules, the maximum between nums[0],
nums[1], and nums[2] is 1.
```

Example 3:

```
Input: n = 3
Output: 2
Explanation: According to the given rules, the maximum between nums[0],
nums[1], nums[2], and nums[3] is 2.
```

Constraints:

- `0 <= n <= 100`

-
  ```
  class Solution:
      def getMaximumGenerated(self, n: int) -> int:
          if n==0:
              return 0
          if n==1:
              return 1
          dp = [0]*(n+1)
          dp[0]=0
          dp[1]=1
          for i in range(2,n+1):
              if i%2==0:
                  dp[i] = dp[i//2]
              else:
                  dp[i] = dp[i//2]+dp[i//2+1]
          return max(dp)
  ```