

获取生成数组中的最大值

给你一个整数 n 。按下述规则生成一个长度为 $n + 1$ 的数组 `nums`：

- `nums[0] = 0`
- `nums[1] = 1`
- 当 $2 \leq 2 * i \leq n$ 时，`nums[2 * i] = nums[i]`
- 当 $2 \leq 2 * i + 1 \leq n$ 时，`nums[2 * i + 1] = nums[i] + nums[i + 1]`

返回生成数组 `nums` 中的 **最大值**。

示例 1：

输入： $n = 7$

输出：3

解释：根据规则：

$$\text{nums}[0] = 0$$

$$\text{nums}[1] = 1$$

$$\text{nums}[(1 * 2) = 2] = \text{nums}[1] = 1$$

$$\text{nums}[(1 * 2) + 1 = 3] = \text{nums}[1] + \text{nums}[2] = 1 + 1 = 2$$

$$\text{nums}[(2 * 2) = 4] = \text{nums}[2] = 1$$

$$\text{nums}[(2 * 2) + 1 = 5] = \text{nums}[2] + \text{nums}[3] = 1 + 2 = 3$$

$$\text{nums}[(3 * 2) = 6] = \text{nums}[3] = 2$$

$$\text{nums}[(3 * 2) + 1 = 7] = \text{nums}[3] + \text{nums}[4] = 2 + 1 = 3$$

因此，`nums = [0,1,1,2,1,3,2,3]`，最大值 3

示例 2：

输入： $n = 2$

输出：1

解释：根据规则，`nums[0]`、`nums[1]` 和 `nums[2]` 之中的最大值是 1

示例 3：

输入：n = 3

输出：2

解释：根据规则，nums[0]、nums[1]、nums[2] 和 nums[3] 之中的最大值是 2

```
int cmp(const void* a,const void* b)
{
    return *(int*)a-*(int*)b;
}
int getMaximumGenerated(int n){
    int* nums=(int*)malloc(sizeof(int)*(n+1));
    for(int i=0;i<n+1;i++)
    {
        if(i==0)
            nums[i]=0;
        if(i==1)
            nums[i]=1;
        if(i%2==0&& i*2>=2)
            nums[i]=nums[i/2];
        if(i%2==1&& i*2+1>=2)
            nums[i]=nums[i/2]+nums[i-i/2];
    }
    qsort(nums,n+1,sizeof(int),cmp);
    return nums[n];
}
```