

We define a harmonious array as an array where the difference between its maximum value and its minimum value is exactly 1.

Given an integer array nums, return the length of its longest harmonious subsequence among all its possible subsequences.

A subsequence of an array is a sequence that can be derived from the array by deleting some or no elements without changing the order of the remaining elements.

Example 1: Input: nums = [1,3,2,2,5,2,3,7] Output: 5

In [3]:

```
1 def findLHS(nums):
2     freq = {}
3     for num in nums:
4         freq[num] = freq.get(num, 0) + 1
5
6     max_length = 0
7     for num in freq:
8         if num+1 in freq:
9             max_length = max(max_length, freq[num] + freq[num+1])
10    return max_length
11
12 nums = [1,3,2,2,5,2,3,7]
13 result = findLHS(nums)
14 print("Maximum length of harmonious subsequence is",result)
```

Maximum length of harmonious subsequence is 5

Question 4 You have a long flowerbed in which some of the plots are planted, and some are not. However, flowers cannot be planted in adjacent plots. Given an integer array flowerbed containing 0's and 1's, where 0 means empty and 1 means not empty, and an integer n, return true if n new flowers can be planted in the flowerbed without violating the no-adjacent-flowers rule and false otherwise.

Example 1: Input: flowerbed = [1,0,0,0,1], n = 1 Output: true

In [2]:

```
1 def canflowerplace(flower,n):
2     if n == 0:
3         return True
4     for i in range(len(flower)):
5         if flower[i] == 0 and (i == 0 or flower[i-1] == 0) and (i == len(flower)-1 or flower[i+1] == 0):
6             flower[i] = 1
7             n = n - 1
8             if n == 0:
9                 return True
10    return False
11
12
13 canflowerplace([1,0,0,0,1],1)
```

Out[2]:

True

```
1 Question 5
2 Given an integer array nums, find three numbers whose product is maximum and return the maximum product.
```

```
3
4 Example 1:
5 Input: nums = [1,2,3]
6 Output: 6
```

In [5]:

```
1 def maximum_product(nums):
2     n = len(nums)
3     nums.sort()
4
5     return max(nums[n-1] * nums[n-2] * nums[n-3], nums[0] * nums[1] * nums[n-1])
6
7 maximum_product([1,2,3])
```

Out[5]:

6

```
1 Question 6
2 Given an array of integers nums which is sorted in ascending order, and an integer
3 target,
4 write a function to search target in nums. If target exists, then return its index.
5 Otherwise,
6 return -1.
7
8 You must write an algorithm with O(log n) runtime complexity.
9
10 Input: nums = [-1,0,3,5,9,12], target = 9
11 Output: 4
```

In [7]:

```
1 def search_ele(nums, target):
2     s,e = 0, len(nums)-1
3
4     while s < e:
5         m = s + (e-s)//2
6         if nums[m] == target:
7             return m
8         elif nums[m] > target:
9             e = m - 1
10        else:
11            s = m + 1
12
13    return -1
14
15
16 nums = [-1,0,3,5,9,12]
17 target = 9
18 result = search_ele(nums,target)
19 print(result)
```

4

```
1 Question 7
2 An array is monotonic if it is either monotone increasing or monotone decreasing.
3
4 An array nums is monotone increasing if for all i <= j, nums[i] <= nums[j]. An array nums
5 is
6 monotone decreasing if for all i <= j, nums[i] >= nums[j].
```

```
7 Given an integer array nums, return true if the given array is monotonic, or false otherwise
```

In [8]:

```
1 def isMonotonic(nums):
2     n = len(nums)
3     incre = decre = True
4
5     for i in range(1,n):
6         if nums[i] < nums[i-1]:
7             incre = False
8         elif nums[i] > nums[i-1]:
9             decre = False
10
11     return incre or decre
12
13
14 isMonotonic([1,2,2,3])
```

Out[8]:

True

```
1 Alice has n candies, where the ith candy is of type candyType[i]. Alice noticed that she
  started to gain weight, so she visited a doctor.
2
3 The doctor advised Alice to only eat n / 2 of the candies she has (n is always even).
  Alice likes her candies very much, and she wants to eat the maximum number of different
  types of candies while still following the doctor's advice.
4
5 Given the integer array candyType of length n, return the maximum number of different
  types of candies she can eat if she only eats n / 2 of them.
6
7 Example 1:
8 Input: candyType = [1,1,2,2,3,3]
9 Output: 3
10
```

In [12]:

```
1 def max_candies(candytype):
2     unique_candies = set(candytype)
3     candy_limit = len(candytype) // 2
4
5     return min(len(unique_candies), candy_limit)
6
7 candyType = [1, 1, 2, 2, 3, 3]
8 print(max_candies(candyType))
```

3

In []:

```
1
```

