*"""  
给定一个整数数组[a1,a2,.....aN] ，N个数,  现在从里面选择若干数使得他们的和最大，同时满足相邻两数不能同时被选， a1和aN首尾两个也认为是相邻的。  
"""***class** Solution:  
 **def** \_\_init\_\_(self, numbers):  
 self.numbers = numbers  
  
 **def** chooseMax(self):  
 self.res = []  
 self.dfs([], self.numbers)  
 **return** self.res  
  
 **def** dfs(self, memo, numbers):  
 **if** sum(memo) > sum(self.res):  
 self.res = sorted(memo)  
  
 **if** len(numbers) == 0:  
 **return  
  
 if** len(numbers) == 1:  
 memo.append(numbers[0])  
 **if** sum(memo) > sum(self.res):  
 self.res = sorted(memo)  
 memo.pop()  
 **return  
  
 if** len(numbers) == 2:  
 memo.append(numbers[0])  
 **if** sum(memo) > sum(self.res):  
 self.res = sorted(memo)  
 memo.pop()  
 memo.append(numbers[1])  
 **if** sum(memo) > sum(self.res):  
 self.res = sorted(memo)  
 memo.pop()  
 **return** *# print("memo", memo)  
 # print("numbers", numbers)  
 # print("res", self.res)* **for** i **in** range(len(self.numbers)):  
 memo.append(self.numbers[i])  
 **if** i == 0:  
 self.dfs(memo, numbers[2:-2])  
 **elif** i == len(numbers) - 1:  
 self.dfs(memo, numbers[1:-3]) *# -3 决定了递归条件有三个* **else**:  
 self.dfs(memo, numbers[:i-1] + numbers[i+2:])  
 memo.pop()  
  
*# 目前存在递归过深的问题  
# 1. 手动模拟递归调用  
# 2. 我似乎在leetcode上看到类似的，好像是使用反推法做的***if** \_\_name\_\_ == **"\_\_main\_\_"**:  
 numbers = [5, 6, 7, 8, 10]  
 sol = Solution(numbers)  
 res = sol.chooseMax()  
 print(**"The final result is: "**, res, sum(res))