README_ A1

一、程序设计思路

每个玩家在每一步都会尝试选择最优策略，即他们会使得自己的得分最大化。这样，玩家 1 想要获胜，他需要每一步得分最大，而且在每一步他都要假设玩家 2 也会选择使得他的得分最小的策略。

canWinHelper 函数使用递归模拟游戏的进行，判断玩家在每个回合的情况下是否有必胜的策略。其参数包括：

nums：整数数组，代表当前剩余的数字。

start 和 end：表示当前数组的起始位置和结束位置。

player1Score 和 player2Score：表示玩家 1 和玩家 2 的得分。

isPlayer1Turn：表示当前是否为玩家 1 的回合。

canWinHelper 首先检查游戏是否结束，若开始位置大于起始位置，则会比较玩家 1 和玩家 2 的得分，返回玩家 1 是否获胜结果。如果游戏未结束，则会根据当前是玩家 1 还是玩家 2 的回合，尝试所有可能的选择，递归调用自己，分别计算玩家的得分，直到游戏结束为止。最后，它会根据玩家 1 和玩家 2 的得分来决定返回值。该算法的核心是处于当前回合的玩家能必胜的条件是：无论该玩家选择头或尾的数字，另一玩家都无法获胜。

canWin 函数是对外接口，调用 canWinHelper 函数来执行递归过程，并返回玩家 1 是否能够必胜的结果，便于使用和阅读。

二、算法分析

输入规模为数组的大小 n。用 T[i][j]表示当前数组剩余第 i 个到第 j 个时，对应玩家是否能必胜。则递推关系可以描述为 T[i][j] = ! T[i+1][j] || ! T[i][j-1]，当 i>j 时，可通过双方得分判断胜利与否。本算法的时间复杂度在最坏情况下为 $O(2^n)$ 。

三、使用说明

输入：数组的元素个数、数组元素。

输出：true/false（表示玩家 1 能否必胜）。

四、测试

1.
```
Please enter the size of array: 3
Please enter the elements: 1 10 3
false
```

2.
```
Please enter the size of array: 4
Please enter the elements: 7 10 6 8
true
```

3.
```
Please enter the size of array: 9
Please enter the elements: 1 3 2 1 2 3 2 3 1
false
```