

# 1423. Maximum Points You Can Obtain from Cards

Difficulty	medium
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Finished	@July 10, 2023
Problem	array
Previously asked company	Amazon Google
website	leetcode

Question:

There are several cards **arranged in a row**, and each card has an associated number of points. The points are given in the integer array `cardPoints`.

In one step, you can take one card from the beginning or from the end of the row. You have to take exactly `k` cards.

Your score is the sum of the points of the cards you have taken.

Given the integer array `cardPoints` and the integer `k`, return the *maximum score* you can obtain.

Example 1:

```
Input: cardPoints = [1,2,3,4,5,6,1], k = 3
Output: 12
Explanation: After the first step, your score will always be 1. However, choosing the rightmost card first will maximize your total score. The optimal strategy is to take the three cards on the right, giving a final score of 1 + 6 + 5 = 12.
```

Example 2:

```
Input: cardPoints = [2,2,2], k = 2
Output: 4
Explanation: Regardless of which two cards you take, your score will always be 4.
```

Example 3:

```
Input: cardPoints = [9,7,7,9,7,7,9], k = 7
Output: 55
Explanation: You have to take all the cards. Your score is the sum of points of all cards.
```

My solution:

Time complexity:  $O(k)$

Space complexity:  $O(1)$

```

class Solution(object):
    def maxScore(self, cardPoints, k):
        n = len(cardPoints)
        res = 0
        for i in range(k):
            res += cardPoints[i]
        temp = res
        dec, inc = 0, 0
        l, r = k-1, n-1
        while l >= 0:
            dec += cardPoints[l]
            inc += cardPoints[r]
            res = max(res, temp-dec+inc)
            l -= 1
            r -= 1
        return res

```

### Optimal solution:

Time complexity:  $O(k)$

Space complexity:  $O(1)$

```

class Solution(object):
    def maxScore(self, cardPoints, k):
        l, r = 0, len(cardPoints) - k
        total = sum(cardPoints[r:])
        res = total

        while r < len(cardPoints):
            total += cardPoints[l] - cardPoints[r]
            res = max(res, total)
            l += 1
            r += 1

        return res

```